

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**ESTUDIO Y COMPARACIÓN DE MÉTODOS MODERNOS Y
MÉTODOS INTELIGENTES PARA LA DETERMINACIÓN DEL
COMPORTAMIENTO EN TIEMPO-FRECUENCIA DE UNA SEÑAL**

**DESARROLLO DE METODOS INTELIGENTES TIEMPO-
FRECUENCIA PARA REDUCIR LA CARGA COMPUTACIONAL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

JONATHAN FERNANDO CHÁVEZ JINEZ

jonathan.chavez@epn.edu.ec

DIRECTOR: MSc. RICARDO XAVIER LLUGSI CANAR

ricardo.llugsi@epn.edu.ec

Quito, agosto 2023

CERTIFICACIONES

Yo, Jonathan Fernando Chávez Jinez declaro que el desarrollo del trabajo de integración curricular aquí expuesto es de mi autoría; no ha sido presentado previamente para grado u obtención de calificación profesional; y, que he consultado las diferentes referencias bibliográficas que se incluyen en este trabajo de integración curricular.

Jonathan Fernando Chávez Jinez

Certifico que el presente trabajo de integración curricular fue desarrollado por Jonathan Fernando Chávez Jinez, bajo mi supervisión.

MSc. Ricardo Llugsí

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JONATHAN CHÁVEZ JINEZ

RICARDO LLUGSI

DEDICATORIA

A mis padres Diego y Martha, a mis hermanos Stephanie y Alexander.

Para Alicita y Humbertito.

AGRADECIMIENTO

Este trabajo aporta al material que se tiene de las clases impartidas por el profesor Robin Álvarez Rueda quien asesoró en prácticamente la totalidad del trabajo. Esperamos que este esfuerzo contribuya a ampliar el alcance de este proyecto de investigación.

Extender un agradecimiento especial al Dr. Robin Álvarez por la gran dedicación y consideración que me ha brindado durante este trabajo de titulación, por la empatía y ayuda en los momentos más complicados.

Agradezco a mis padres por todo el apoyo y cariño recibido ya que sin ellos esto no hubiese sido posible lograr todo esto. A mis hermanos por su compañía y cariño que me han brindado en los buenos y malos momentos.

A mis abuelitos, tíos y primos que han sido una ayuda, apoyo y soporte en muchos momentos de este trayecto.

A mi enamorada por el apoyo y el cariño incondicional que tuve y sigo teniendo de ella.

Finalmente agradezco al Dr. Ricardo Llugsí por la ayuda recibida en este Trabajo de Integración Curricular.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS (considerar mi sugerencia en el índice)	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO.....	3
1.4.1 GENERACIÓN DE SEÑAL ARTIFICIAL MULTICOMPONENTE.....	3
1.4.1.1 Resolución en frecuencia.	3
1.4.1.2 Resolución en amplitud.....	3
1.4.1.3 Resolución en tiempo.....	4
1.4.2 LA TRANSFORMADA DE FOURIER DE TIEMPO CORTO (STFT)	4
1.4.3 FILTRO SAVITZKY-GOLAY	6
1.4.4 OTRAS TRANSFORMADAS	6
1.4.5 IDEA PARA REDUCIR LA CARGA COMPUTACIONAL.....	7
2 METODOLOGÍA.....	8
2.1 RESULTADOS GENERACIÓN DE SEÑAL ARTIFICIAL MULTICOMPONENTE.....	8
2.2 DIAGRAMA TF IDEAL	13
2.3 FUNCIÓN SPECTROGRAM Y STFT EN MATLAB	15
2.4 FUNCIÓN FINDPEAKS Y TIC TOC.....	16
2.5 FUNCIÓN FILTRO SAVITZKY-GOLAY	17
2.6 IMPLEMENTACIÓN DEL ALGORITMO PROPUESTO	18
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	26
3.1 RESULTADOS.....	26
3.1.1 Prueba 1.	26
3.1.2 Prueba 2.	33
3.1.3 COMPARACIÓN RESULTADOS	37

3.2	CONCLUSIONES.....	39
3.3	RECOMENDACIONES	40
4	REFERENCIAS BIBLIOGRÁFICAS	40
5	ANEXOS.....	43
	ANEXO I.....	44
	ANEXO II.....	55

RESUMEN

En el presente estudio se mostró los resultados obtenidos en el desarrollo de Métodos “Inteligentes” para reducir la carga computacional, el proceso con el cual se pueden obtener las diferentes frecuencias de interés y la interpretación de la información obtenida respecto al tiempo de procesamiento de cada una de las etapas. Se utilizó como base una señal compuesta por varias componentes independientes, con frecuencias cada vez más cercanas, que permitan evaluar su Resolución en Frecuencia, y amplitudes cada vez más pequeñas, que permitan evaluar la capacidad de detección de amplitudes muy pequeñas, y con duraciones determinadas, que permitan evaluar la Resolución de Tiempo, para cualquier método Tiempo-Frecuencia.

Para este estudio se utilizó el diagrama de tiempo frecuencia, gráficos de espectros utilizando las ventanas de 1-Hann, que genera un ancho de lóbulo principal más angosto que la ventana rectangular, y Blackman Harris, que genera lóbulos secundarios muy pequeños del orden de 92 dB por debajo del pico del lóbulo principal.

En este estudio se realizó 2 pruebas y 6 escenarios con distintas señales artificiales para mostrar y comparar la capacidad de los métodos frente a las características antes mencionadas. El primer escenario tiene parámetros estándar, amplitudes son igual a uno y las componentes de frecuencia se encuentran separadas considerablemente. En el segundo escenario se evaluó las características de Resolución en Frecuencia. El tercer escenario se evaluó las características de Detección de Amplitudes muy pequeñas. En el cuarto escenario se evaluaron las características de Resolución en Frecuencia y Detección de Amplitudes pequeñas. El quinto escenario evaluó la Resolución de Tiempo. Finalmente, el sexto escenario evaluó el comportamiento del sistema al adicionar ruido blanco gaussiano.

PALABRAS CLAVE: Métodos Inteligentes, Blackman Harris, 1-Hann, tiempo-frecuencia, espectro de frecuencia.

ABSTRACT

This study shows the results obtained in the development of "Intelligent" methods to reduce the computational load. The process with which the different frequencies of interest can be obtained, and the interpretation of the information obtained with respect to the processing time of each of the stages.

A signal composed of several independent components is used as a basis, with increasingly closer frequencies, which allow evaluating its Frequency Resolution, and increasingly smaller amplitudes, which allow evaluating the detection capacity of very small amplitudes, and with determined durations, which allow evaluating the Time Resolution, for any Time-Frequency method.

For this study we will use the time-frequency diagram, spectrum plots using the 1-Hann window, which generates a narrower main lobe width than the rectangular window, and Blackman Harris, which generates very small secondary lobes of the order of 92 dB below the peak of the main lobe.

In this study, 6 tests will be performed with different artificial signals to show and compare the capability of the methods with the mentioned characteristics.

In this study, 2 tests and 6 scenarios with different artificial signals will be performed to show and compare the capability of the methods against the mentioned characteristics. The first scenario has standard parameters, the amplitudes are equal to one and the frequency components are considerably separated. The second scenario evaluates the frequency resolution characteristics. The third scenario evaluates the very small amplitude detection characteristics. The fourth scenario evaluates the Frequency Resolution and Small Amplitude Detection characteristics. The fifth scenario evaluates the temporal resolution. Finally, the sixth scenario evaluates the system behavior with the addition of white Gaussian noise.

KEYWORDS: Intelligent Methods, Blackman Harris, 1-Hann, time-frequency, frequency spectrum.

1 INTRODUCCIÓN

Los fenómenos presentes en la naturaleza están representados por señales formadas por varias componentes que tienen diferentes frecuencias, amplitudes y duraciones. Como se conoce, las diferentes técnicas para estimación espectral destacan el problema de no contener información respecto al dominio temporal por lo cual solamente se conocerán las componentes de frecuencia que tiene una determinada señal, pero no se tiene información acerca del momento en que ocurre cada una de ellas.

Para tratar de superar esta problemática, durante la historia aparecieron varios métodos que tratan de reducir dicho compromiso; pero, la complejidad y un sinnúmero de métodos existentes, elegir el enfoque adecuado se convirtió en un desafío para investigadores en diversas disciplinas, un uso inapropiado de estas metodologías o la falta de comprensión de sus limitantes puede tener consecuencias severas a la hora de analizar los resultados.

Con el transcurso del tiempo, se han llevado a cabo varios estudios comparativos entre los distintos métodos o enfoques tradicionales usados en el análisis de señales en el dominio de tiempo-frecuencia [1], [2], [3], [4], [5]; no obstante, no se ha hallado un artículo que realice una confrontación de las cualidades significativas como la resolución en tiempo, resolución en frecuencia, capacidad de detección y carga computacional.

Si bien estos últimos métodos obtienen mejores resultados que el espectrograma, varios de estos implican un alto coste computacional y por ello es importante también determinar métodos alternativos que no solamente den buenos resultados, sino que además reduzcan considerablemente dicho coste computacional.

Aparecen varias ideas que se fundamentan en dividir por un lado el análisis espectral, en donde ya se tienen todas las componentes que conforman la señal en análisis, y por otro lado realizar un análisis de la duración de cada una de esas componentes, obteniéndose así un nuevo paradigma que rompe con las ideas tradicionales y que justamente conllevan aquella tremenda carga computacional.

Con este nuevo paradigma será posible reducir el tiempo de cómputo tratando de mantener la calidad de los resultados sobre la evolución de las componentes de frecuencia de la señal.

1.1 OBJETIVO GENERAL

Reducir la carga computacional de un algoritmo TF de modo que se pueda mostrar el comportamiento de una señal tanto en frecuencias como en tiempo

1.2 OBJETIVOS ESPECÍFICOS (considerar mi sugerencia en el índice)

1. Comprender la causa de la carga computacional de los algoritmos TF tradicionales
2. Idear un nuevo algoritmo que permita reducir dicha carga computacional
3. Realizar pruebas para demostrar la reducción de dicha carga computacional

1.3 ALCANCE

Utilizando el software Matlab R2020a, se implementará señales compuestas por diferentes componentes con características definidas en duración, amplitud y frecuencia. En base a esto se podrá realizar las respectivas pruebas y análisis en los dominios de tiempo y frecuencia, las cualidades de detección para los diferentes algoritmos basados en la resolución en dominio temporal y dominio de la frecuencia.

Realizar un diagrama tiempo frecuencia ideal para tener una referencia y poder comparar el desempeño de los métodos inteligentes utilizando diferentes parámetros para su análisis.

Para este análisis se empleará funciones desarrolladas y funciones propias de Matlab.

Se realizarán pruebas con varias señales generadas artificialmente por nosotros, de este modo se evalúan las características de Resolución en Frecuencia, Resolución de Amplitud y Resolución en tiempo.

Se realizará una comparación de los resultados obtenidos en cada una de las diferentes pruebas para lo cual se plantea realizar seis pruebas, cada una de ellas usando características específicas para demostrar las características antes citadas.

Para finalizar, se mostrará el comportamiento únicamente de las componentes más relevantes para evidenciar las características de las pruebas que se han planteado, el resto de las componentes serán desechadas, lo cual permitirá reducir la carga computacional.

1.4 MARCO TEÓRICO

1.4.1 GENERACIÓN DE SEÑAL ARTIFICIAL MULTICOMPONENTE

En muchas ramas correspondientes a ciencias aplicadas, se tienen varios tipos de representaciones para una señal física obtenida mediante diferentes formas, estas señales pueden ser representadas mediante una función en el dominio temporal $f(t)$ o en el dominio de la frecuencia $F(\omega)$. Para esto se utiliza la Transformada de Fourier [6].

Las señales también pueden ser creadas artificialmente mediante software y pueden ser multicomponentes. Las señales multicomponentes pueden ser caracterizadas de diferentes formas, pero en este trabajo se tendrán 3 consideraciones específicas las cuales son: Resolución en Frecuencia, Resolución en Amplitud y Resolución en Tiempo.

1.4.1.1 Resolución en frecuencia.

La Resolución en Frecuencia se caracteriza en hallar diferentes componentes que conforman la señal artificial generada. Dichas componentes de frecuencia pueden estar separadas considerablemente o muy cercanas entre sí.

Para visualizar dicha característica se estableció el acercamiento de dos pares de componentes. En la Tabla 1.1 se presentan los pares para estudiar el comportamiento del algoritmo TF.

Tabla 1.1: Acercamiento de componentes para mostrar la resolución en frecuencia.

ESCENARIO			
INICIAL		CAMBIO	
Componente	Frecuencia	Componente	Frecuencia
C2	300 [Hz]	C2	300 [Hz]
C3	400 [Hz]	C3	304 [Hz]
C6	800 [Hz]	C6	800 [Hz]
C7	900 [Hz]	C7	810 [Hz]

1.4.1.2 Resolución en amplitud.

La Resolución en Amplitud se caracteriza por encontrar las componentes con amplitudes normales y muy pequeñas que conforman la señal artificial. La amplitud de las componentes a estudiar tendrá una disminución paulatina, presentada en la Tabla 1.2, con el fin de mostrar el desempeño del método TF.

Tabla 1.2: Variación de amplitud de las componentes que conforman la señal artificial a utilizar.

ESCENARIO			
INICIAL		CAMBIO	
Componente	Amplitud	Componente	Amplitud
C1	1	C1	1
C2	1	C2	1
C3	1	C3	0.5
C4	1	C4	0.1
C5	1	C5	0.2
C6	1	C6	0.4
C7	1	C7	0.01

1.4.1.3 Resolución en tiempo.

Cada una de las componentes que conforman la señal artificial creada tiene una duración que puede ser igual o diferente entre las componentes para lo cual se establece una variación adicional con el fin de observar el comportamiento del algoritmo.

En la Tabla 1.3 se muestra como las componentes tienen valor inicial y final en el tiempo, los cuales serán modificados para su estudio.

Tabla 1.3: Cambios de intervalos de tiempo para las componentes a estudiar.

ESCENARIO			
INICIAL		CAMBIO	
Componente	Duración [s]	Componente	Duración [s]
C1	[0 3]	C1	[0 2]
C2	[2 4]	C2	[5 6]
C3	[2 3]	C3	[5 8]
C4	[4 6]	C4	[2 4]
C5	[1 2]	C5	[1 3]
C6	[3 4]	C6	[2 6]
C7	[4 5]	C7	[1 3]

1.4.2 LA TRANSFORMADA DE FOURIER DE TIEMPO CORTO (STFT)

La Transformada de Fourier es una herramienta utilizada para encontrar información sobre cómo está constituida la señal a analizar respecto a las diferentes frecuencias que la componen [7] [8].

Dicha transformada tiene la capacidad de encontrar las componentes en frecuencia, pero sin presentar características en tiempo, para lo cual se diseña la STFT (*Short Time Fourier Transform*, Transformada de Fourier de Tiempo Corto) [7] [8].

STFT es la versión mejorada o perfeccionada de la Transformada de Fourier, el gran cambio y aporte obtenido por esta herramienta es ubicar los instantes de tiempo en los que aparecen, o son encontradas, las diferentes componentes de frecuencia. De esta manera se observan periodogramas tridimensionales teniendo como ejes el tiempo, la frecuencia y la energía que proporciona cada una de las componentes [7] [8].

La Transformada de Fourier de Tiempo Corto es el resultado de aplicar en cortos intervalos de tiempo la Transformada de Fourier, en otras palabras, consiste en multiplicar una señal de entrada por una ventana definida la cual se traslada en el tiempo.

Como resultado de esta operación se obtienen las componentes de frecuencia dentro de la ventana en cada una de las transiciones [7] [8].

En la ecuación 1 se encuentra definida la Transformada de Fourier de Tiempo Corto para señales continuas [7].

$$STFT(\tau, f) = \int_0^t [y(t) * w(t - \tau)] * e^{i*2*\pi*f*t} dt \quad (1)$$

Donde:

$y(t)$: Señal definida en t.

$w(t - \tau)$: Función ventana con tiempo de retraso igual a τ .

f : Frecuencia

t : Tiempo

τ' : Tiempo del centro de la ventana $w(t)$

$STFT(\tau, f)$: Valor de la Transformada de Fourier de Tiempo Corto para un centroide de ventana τ y una frecuencia f específica.

Mientras que la expresión de la Transformada de Fourier de Tiempo Corto para señales discretas se define en la ecuación 2 [7].

$$STFT(\tau, f) = \sum_{f=0}^{N-1} \sum_{\tau'=0}^{N-1} [y(t) * w(t - \tau)] * e^{i*2*\pi*f*t} dt \quad (2)$$

Donde:

N : Numero de datos de la señal.

1.4.3 FILTRO SAVITZKY-GOLAY

Savitzky Golay es un filtro pasa bajos el cual utiliza la aproximación polinomial de mínimos cuadrados locales. Se han encontrado algunas propiedades al utilizarlo en aplicaciones como procesamiento de señales de electrocardiograma, electroencefalogramas [17].

Este filtro se basa en un análisis de regresión polinómica de orden p con un número mínimo de $2n + 1$ puntos equidistantes a través de una ventana. El principio de $2n + 1$ conjunto de puntos igualmente espaciados $(p_n, \dots, p_n, \dots, p_n)$ es una representación de diferentes muestras del polinomio de grado p [18].

Matemáticamente, el filtro se expresa [18]:

$$\hat{T} = \frac{1}{n} \sum_{i=-m}^m C_i T_{j+1} \quad (3)$$

Donde:

T : valor original en tiempo.

m : ventana del filtro.

n : número entero de convolución.

\hat{T} : valor estimado.

C_i : coeficiente par del i -ésimo valor del filtro.

1.4.4 OTRAS TRANSFORMADAS

Existen varios métodos para el análisis en tiempo-frecuencia que han sido desarrollados en los últimos años. El objetivo de estos desarrollos es el complementar, los algoritmos matemáticos para el análisis de señales en frecuencia, con métodos óptimos con el propósito de investigar y examinar en el contexto temporal [9].

Uno de estos métodos es la Transformada de Gabor, que constituye una variante particular de la Transformada de Fourier (STFT). Se caracteriza por usar una ventana deslizante Gaussiana ya que representa una distribución normal y matemáticamente, se puede obtener la inversa de la Transformada de Gabor [9].

Para analizar una señal $x(t)$ se utiliza la ventana de Gabor $g(t)$ en un punto, la acción realizada implica la multiplicación de la señal con la función de ventana trasladada τ y a dicho producto se aplica la *Transformada de Fourier* [19].

Una vez hecho esto se obtiene el valor:

$$G(\tau, \omega) = \int_{-\infty}^{+\infty} x(t)g(t - \tau)e^{-2\pi i t \omega} dt \quad (4)$$

En la ecuación 4 se tienen las siguientes variables [19]:

ω : frecuencia.

$x(t)$: señal de datos.

τ : instante de tiempo.

La Reasignada de Fourier calcula las estimaciones espectrales afinadas en las diferentes frecuencias que componen la señal de estudio. Normalmente se localiza las componentes en el centro geométrico de la ventana utilizada, pero en este caso se las registra al centro de gravedad de la distribución espectral [9].

Una forma de analizar el principio de la reasignación es partir de la causa práctica de la mala localización obtenida con el espectrograma. Consiste en analizar la energía de la señal mediante una ventana tiempo-frecuencia, teniendo desplazamiento en el dominio de tiempo y en el dominio de la frecuencia [20].

Otro método es la Transformada Sincronizada de Fourier, el cual tiene como objetivo afinar la representación en el dominio de tiempo y frecuencia, mientras sigue siendo invertible.

1.4.5 IDEA PARA REDUCIR LA CARGA COMPUTACIONAL

Para reducir la carga computacional de diferentes procesos y análisis se deben considerar variables tanto de hardware y software (optimizando códigos previamente desarrollados con base a la necesidad del sistema).

Algunas formas para la reducción de carga computacional son:

1. Análisis del procedimiento: Utilizar técnicas para mostrar el comportamiento de la señal en las frecuencias que son de interés, descartando el resto de las frecuencias secundarias sin perder la información crítica. Valioso para el análisis en tiempo real de información y mostrar graficas de datos o comportamiento del algoritmo.

2. Optimización de algoritmos: Ajustar y revisar los algoritmos utilizados y desarrollados para mejorar la eficiencia respecto al tiempo y recursos utilizados en el procesamiento de señales.
3. Distribución: Dividir las tareas simultaneas y ejecutarlas en múltiples núcleos del CPU. Esto aumenta la velocidad de procesamiento y reducir el tiempo total necesario para poder completar la ejecución del algoritmo y la presentación de las diferentes graficas a analizar.

Teniendo en cuenta estas ideas, se optimiza la ejecución de los algoritmos para análisis de señales y de esta manera, poder estudiar señales mucho más complejas, con mayor duración en el tiempo y complejidad respecto a las frecuencias que componen la señal a ser analizada.

La reducción de carga computacional mediante métodos inteligentes implica una combinación de optimización algorítmica, uso inteligente de recursos y métodos de automatización de aprendizaje para aumentar la efectividad de los sistemas y procedimientos.

2 METODOLOGÍA

En este capítulo se mostrará el proceso seguido para la obtención, muestra y comparación de los métodos tradicionales Tiempo-Frecuencia vs el algoritmo diseñado para disminuir la carga computacional considerablemente.

2.1 RESULTADOS GENERACIÓN DE SEÑAL ARTIFICIAL MULTICOMPONENTE

Para comenzar con el análisis se debe realizar la gráfica compuesta por varias componentes. Estas componentes tienen las características definidas en frecuencia, amplitud y duración según se muestra en la Tabla 2.1.

Tabla 2.1: Características totales de las componentes que conforman la señal artificial.

Escenarios	Componente	Frecuencia	Amplitud	Duración [s]
1. General	C1	200 [Hz]	1	[0 3]
	C2	300 [Hz]	1	[2 4]
	C3	400 [Hz]	1	[2 3]
	C4	600 [Hz]	1	[4 6]
	C5	700 [Hz]	1	[1 2]
	C6	800 [Hz]	1	[3 4]

	C7	900 [Hz]	1	[4 5]
2. Resolución en frecuencia	C1	200 [Hz]	1	[0 3]
	C2	300 [Hz]	1	[2 4]
	C3	304 [Hz]	1	[2 3]
	C4	600 [Hz]	1	[4 6]
	C5	700 [Hz]	1	[1 2]
	C6	800 [Hz]	1	[3 4]
	C7	810 [Hz]	1	[4 5]
3. Resolución en amplitud	C1	200 [Hz]	1	[0 3]
	C2	300 [Hz]	1	[2 4]
	C3	400 [Hz]	0.5	[2 3]
	C4	600 [Hz]	0.1	[4 6]
	C5	700 [Hz]	0.2	[1 2]
	C6	800 [Hz]	0.4	[3 4]
	C7	900 [Hz]	0.01	[4 5]
4. Resolución en frecuencia y amplitud	C1	200 [Hz]	1	[0 3]
	C2	300 [Hz]	1	[2 4]
	C3	304 [Hz]	0.5	[2 3]
	C4	600 [Hz]	0.1	[4 6]
	C5	700 [Hz]	0.2	[1 2]
	C6	800 [Hz]	0.4	[3 4]
	C7	810 [Hz]	0.03	[4 5]
5. Resolución en tiempo	C1	200 [Hz]	1	[0 2]
	C2	300 [Hz]	1	[5 6]
	C3	400 [Hz]	1	[5 8]
	C4	600 [Hz]	1	[2 4]
	C5	700 [Hz]	1	[1 3]
	C6	800 [Hz]	1	[2 6]
	C7	900 [Hz]	1	[1 3]
6. General con ruido AWGN - 25 [dB]	C1	200 [Hz]	1	[0 3]
	C2	300 [Hz]	1	[2 4]
	C3	400 [Hz]	1	[2 3]
	C4	600 [Hz]	1	[4 6]
	C5	700 [Hz]	1	[1 2]
	C6	800 [Hz]	1	[3 4]
	C7	900 [Hz]	1	[4 5]

La señal total “*y_{ts}*” está conformada por 7 componentes, como presenta la Figura 2.1, con un valor de frecuencia, amplitud y duración determinada. Los valores de frecuencia utilizados son asignados a un vector “*F_{tot}*” para poder obtener la frecuencia mayor entre ellas. Dicho valor es utilizado para obtener la frecuencia de muestreo “*F_s*”, la cual es igual

a 10 veces el valor de la frecuencia máxima encontrada en el vector de frecuencias. En adición a esto el tiempo de muestreo “Ts” es igual a la inversa de la frecuencia de muestreo.

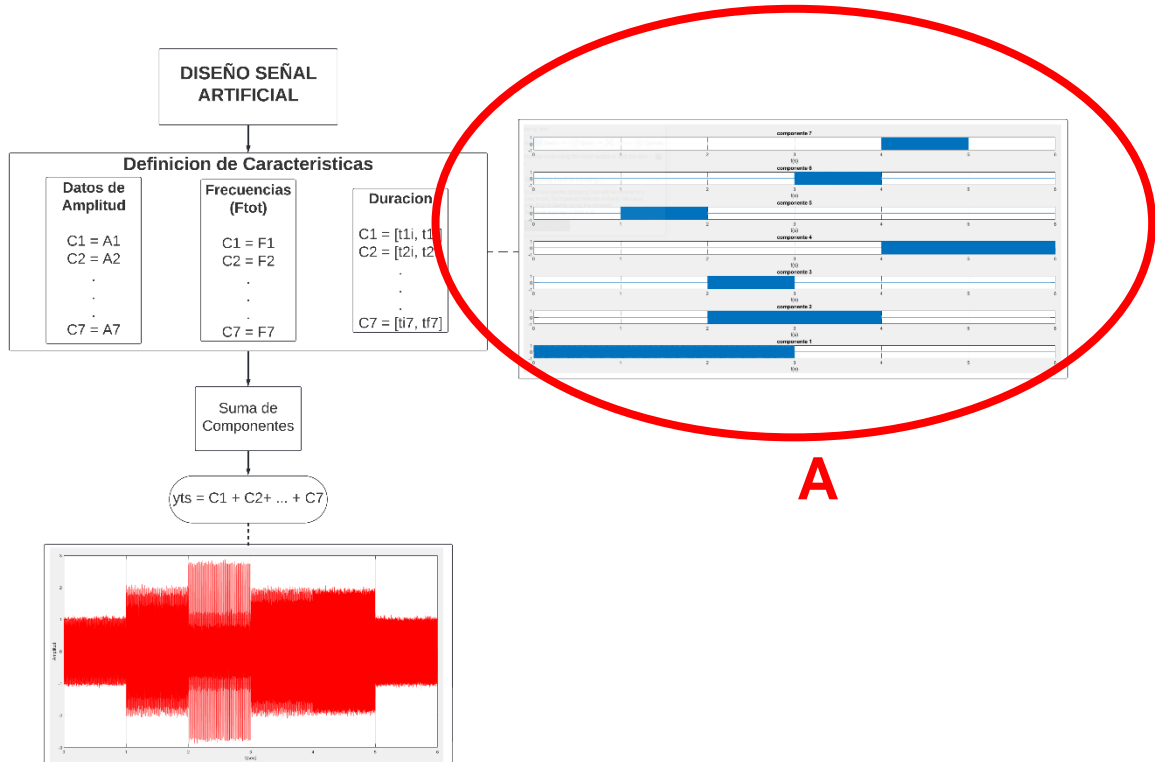


Figura 2.1: Diseño de señal artificial para el estudio.

En la Figura 2.2 se observa las características asignadas a cada componente que conforma la señal artificial.

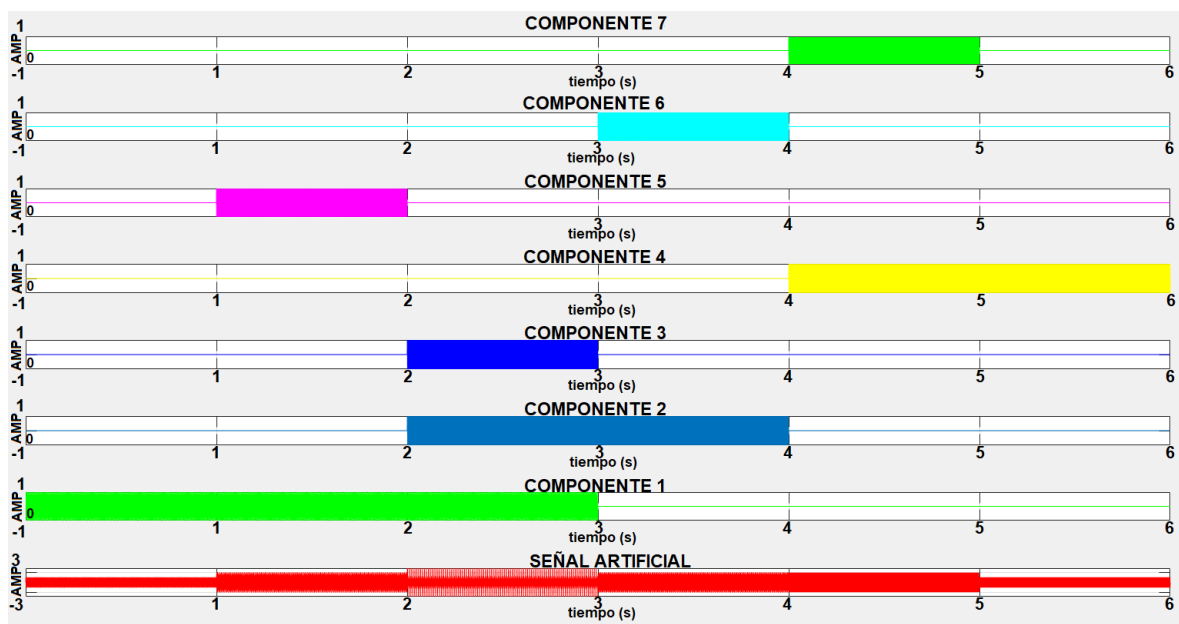


Figura 2.2: Ampliación A correspondiente a la Figura 1. Componentes utilizadas para el diseño de la señal artificial.

Ahora presentamos el fragmento de código empleado para crear la señal artificial de múltiples componentes utilizada en la evaluación de enfoques inteligentes.

```
%% 1. Señal Multicomponente:
% Los datos deben irse descomentando dependiendo del
escenario a estudiar:

% 1.1 Frecuencias de componentes, ordenadas
ascendentemente:
f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 1
% f1=200; f2=300; f3=304; f4=600; f5=700; f6=800;
f7=810; % Escenario 2
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 3
% f1=200; f2=300; f3=304; f4=600; f5=700; f6=800;
f7=810; % Escenario 4
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 5
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 6
Ftot=[f1 f2 f3 f4 f5 f6 f7]; % Vector de frecuencias
fmax=max(Ftot);

% % 1.2 Amplitudes:
a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; % Escenario
1
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; %
Escenario 2
% a1=1; a2=0.8; a3=0.5; a4=0.1; a5=0.2; a6=0.4;
a7=0.01; % Escenario 3
% a1=1; a2=1; a3=0.5; a4=0.1; a5=0.2; a6=0.4; a7=0.03;
% Escenario 4
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; %
Escenario 5
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; %
Escenario 6
Atot=[a1 a2 a3 a4 a5 a6 a7]; % Vector de amplitudes

% 1.3 Lapso de cada componente:
t1=[0 3]; t2=[2 4]; t3=[2 3]; t4=[4 6]; t5=[1 2];
t6=[3 4]; t7=[4 5]; % Escenario 1, 2, 3, 4, 6
% t1=[0 2]; t2=[5 6]; t3=[5 8]; t4=[2 4]; t5=[1 3];
t6=[2 6]; t7=[1 3]; % Escenario 5
```

```

tT = [t1,t2,t3,t4,t5,t6,t7];

%% 2. Creación de señal compuesta:
Fs=10*fmax; % FSmin >= 2*Fmax Nyquist
Ts=1/Fs;
duracion = max(tT); % duración total componentes
t=0:Ts:duracion;

% Generacion de componentes por intervalos
comp1=a1*sin(2*pi*f1*t);
comp2=a2*sin(2*pi*f2*t);
comp3=a3*sin(2*pi*f3*t);
comp4=a4*sin(2*pi*f4*t);
comp5=a5*sin(2*pi*f5*t);
comp6=a6*sin(2*pi*f6*t);
comp7=a7*sin(2*pi*f7*t);

% ----- Escenarios 1,2,3,4,6 -----
-%
comp1=(t<=3) .* (comp1);
comp2=( (t>=2) & (t<=4) ) .* (comp2);
comp3=( (t>2) & (t<=3) ) .* (comp3);
comp4=( (t>=4) & (t<=6) ) .* (comp4);
comp5=( (t>=1) & (t<=2) ) .* (comp5);
comp6=( (t>=3) & (t<=4) ) .* (comp6);
comp7=( (t>=4) & (t<=5) ) .* (comp7);
% -----%

% ----- Escenario 5 -----%
% comp1=(t<=2) .* (comp1);
% comp2=( (t>=5) & (t<=6) ) .* (comp2);
% comp3=( (t>5) & (t<=8) ) .* (comp3);
% comp4=( (t>=2) & (t<=4) ) .* (comp4);
% comp5=( (t>=1) & (t<=3) ) .* (comp5);
% comp6=( (t>=2) & (t<=6) ) .* (comp6);
% comp7=( (t>=1) & (t<=3) ) .* (comp7);
% -----%

% Se muestran las diferentes componentes para la señal
artificial, ordenada
% ascendientemente por la frecuencia:
figure(1)
subplot(8,1,1);plot(t,comp7,'g');title('Señal 7');
xlabel('t(s)');grid on;

```

```

subplot(8,1,2);plot(t,comp6,'c');title('Señal 6');
xlabel('t(s)');grid on;
subplot(8,1,3);plot(t,comp5,'m');title('Señal 5');
xlabel('t(s)');grid on;
subplot(8,1,4);plot(t,comp4,'y');title('Señal 4');
xlabel('t(s)');grid on;
subplot(8,1,5);plot(t,comp3,'b');title('Señal 3');
xlabel('t(s)');grid on;
subplot(8,1,6);plot(t,comp2);title('Señal 2');
xlabel('t(s)');grid on;
subplot(8,1,7);plot(t,comp1,'g');title('Señal 1');
xlabel('t(s)');grid on;

% Agregado de componentes para obtener la señal
completa.
yts = comp1+comp2+comp3+comp4+comp5+comp6+comp7;

% ----- Escenario 6 -----
% ----- %
% SNR = 25;
% yts = awgn(yts,SNR,'measured');
% -----
% ----- %

subplot(8,1,8);
plot(t,yts,'r');
title('Señal Artificial');
xlabel('t(s)');
grid on;

```

2.2 DIAGRAMA TF IDEAL

El diagrama Tiempo-Frecuencia es una representación del comportamiento de diferentes componentes de frecuencia a través del tiempo. La Figura 2.1 muestra el proceder de una señal artificial en el tiempo y para complementar esta información se tiene el diagrama presentado en la Figura 2.3.

Este diagrama muestra cual es la duración y el lapso (tiempo inicial – tiempo final) de cada una de las componentes en frecuencia que componen una señal.

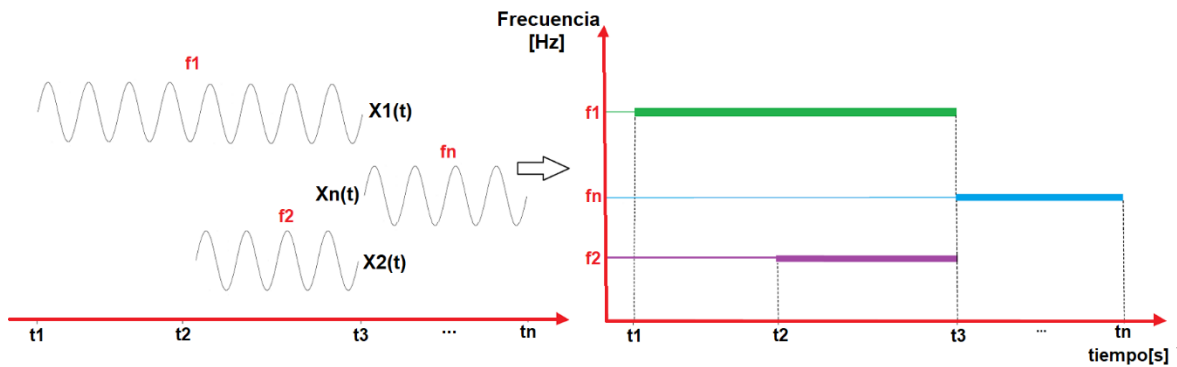


Figura 2.3: Proceso para obtener el diagrama Tiempo-Frecuencia.

La Figura 2.4 muestra el comportamiento de las siete componentes que componen la señal artificial diseñada. Estos valores corresponden al escenario 1 establecido en la Tabla 2.1.

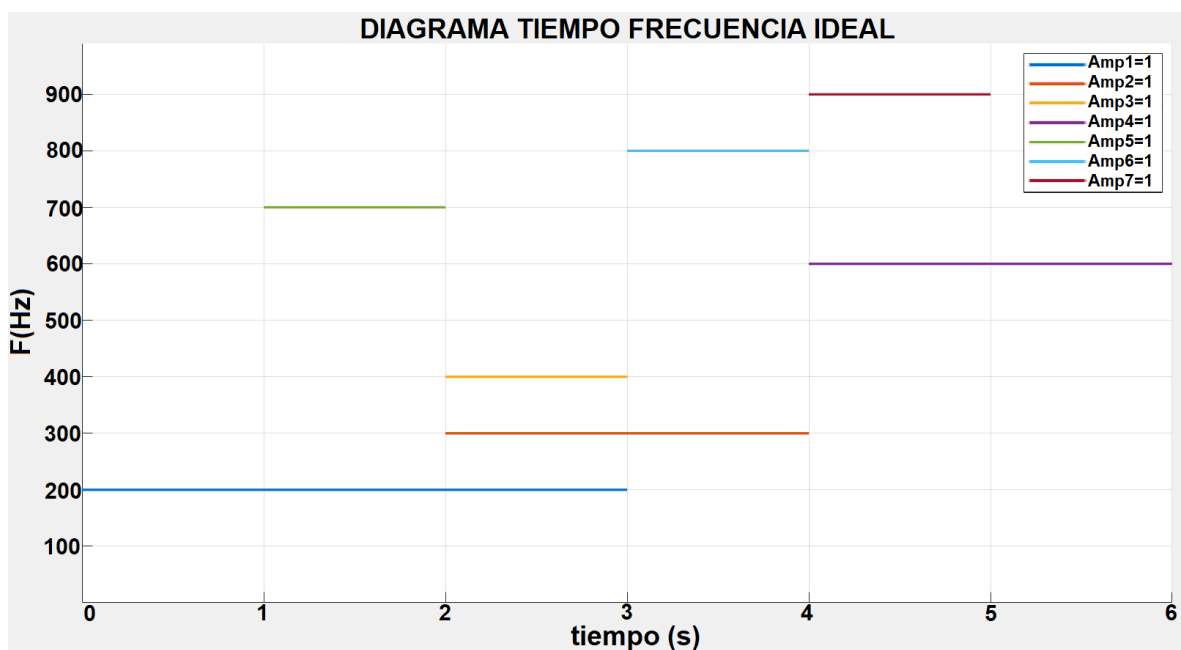


Figura 2.4: Diagrama Tiempo-Frecuencia Ideal.

El siguiente código muestra los pasos utilizados para poder obtener el diagrama Tiempo - Frecuencia ideal, utilizado como referencia para los resultados finales usando métodos inteligentes.

```
% Presentacion del diagrama tiempo-frecuencia ideal,
mostrando la duracion y la frecuencia previamente
establecidas.
figure(2)
hold on
plot (t1,[f1 f1], 'linewidth', 2)
plot (t2,[f2 f2], 'linewidth', 2)
```

```

plot (t3,[f3 f3],'linewidth',2)
plot (t4,[f4 f4],'linewidth',2)
plot (t5,[f5 f5],'linewidth',2)
plot (t6,[f6 f6],'linewidth',2)
plot (t7,[f7 f7],'linewidth',2)
grid on
title('DIAGRAMA TIEMPO-FRECUENCIA IDEAL')
xlabel('t(sec)')
ylabel('F(Hz)')
axis([0 duracion 0 fmax*1.1])

legend('Amp.1: 1','Amp.2: 1','Amp.3: 1','Amp.4:
1','Amp.5: 1','Amp.6: 1','Amp.7: 1') % Escenario 1

```

2.3 FUNCIÓN SPECTROGRAM Y STFT EN MATLAB

La Transformada de Fourier de Tiempo Corto se emplea para examinar cómo la distribución de frecuencias evoluciona a lo largo del tiempo.

```
[s, f, t] = stft(x, fs)
```

Tabla 2.2: Parámetros de entrada y salida Transformada de Fourier de Tiempo Corto [11]

PARÁMETRO	DESCRIPCIÓN
fs	Frecuencia de muestreo
x	Señal artificial diseñada para análisis.
s	STFT
f	Vector de frecuencias obtenidas por la STFT
t	Vector de tiempos obtenidos por la STFT

La función *spectrogram* utiliza como base la Transformada de Fourier de Tiempo Corto. Existen varias formas de obtener resultados dependiendo de los datos de entrada de la función.


```
[s, f, t] = spectrogram(x, window, noverlap, nfft, fs)
```

Tabla 2.3: Parámetros de entrada y salida función *spectrogram* [12]

PARÁMETRO	DESCRIPCIÓN
x	Señal artificial diseñada para análisis.
window	División de la señal en segmentos y disponerlos en ventanas
noverlap	Muestras de solapamiento entre los segmentos continuos
nfft	Utiliza puntos de muestras para el cálculo de la transformada discreta de Fourier
fs	Frecuencia de muestreo (10 veces la fmax obtenida)
s	Estimación de potencia en tiempo-frecuencia
f	Vector de frecuencias cíclicas a una tasa de muestreo fs
t	Vector instantes de tiempo en los que se calcula la STFT

2.4 FUNCIÓN FINDPEAKS Y TIC TOC

“*Findpeaks*” es una función de Matlab que entrega diferentes vectores de valores como la potencia y la frecuencia. La Tabla 2.4 presenta las variables de entrada utilizadas en el desarrollo del programa [13].

```
[power, freq]=findpeaks(p, f);
```

Tabla 2.4: Parámetros de entrada *findpeaks* [13].

PARÁMETRO	DESCRIPCIÓN
p	Vector de potencias de un espectro
f	Vector de frecuencia encontrado con la función spectrogram.

MinPeakProminence	Mínima prominencia del pico
MinPeakWidth	Anchura mínima del pico
MinPeakDistance	Separación mínima de los picos

La función *tic toc* son operaciones para medir el tiempo. TIC registra el valor actual de tiempo, mientras que el TOC registra el valor final del tiempo. Una vez realizado esto, se calcula la diferencia de tiempo obteniendo la duración del algoritmo deseado [14].

```
tic
timerVal = tic

tic
timerVal = toc
```

2.5 FUNCIÓN FILTRO SAVITZKY-GOLAY

Este filtro se utiliza para poder suavizar señales con ruido considerable. Este filtro es denominado como polinomial y es óptimo para minimizar el error de mínimos cuadrados [15] [16].

```
y = sgolayfilt(x,order,framelen)
```

Tabla 2.5: Parámetros de entrada filtro *sgolay* [15].

PARÁMETRO	DESCRIPCIÓN
x	Vector datos de entrada
order	Orden polinomial
framelen	Longitud del cuadro de datos.

El filtro *sgolay* se utiliza generalmente en procesos de análisis y procesamiento de señales y espectroscopia. Reduce el ruido de altas frecuencias en una señal gracias a sus características de suavizado reduciendo la señal en bajas frecuencias [16].

2.6 IMPLEMENTACIÓN DEL ALGORITMO PROPUESTO

Para comenzar con este trabajo, se necesita desarrollar la señal artificial multicomponente la cual constará de 7 componentes con diferentes valores de amplitud, frecuencia y lapso. Estos valores se encuentran planteados en la Tabla 2.1, la cual muestra los diferentes escenarios de análisis de carga computacional, dicha señal computacional se implementa como se explica en el apartado 2.1.

El equipo utilizado para la experimentación es una laptop con el software Matlab con versión R2020a. Sus características son: Windows 10 Home, procesador Intel CORE i7 8th Gen @ 1.80GHz, RAM 16 GB.

La señal multicomponente es utilizada para poder obtener el análisis espectral usando ventanas con características definidas. Las ventanas utilizadas para este desarrollo son 1-Hann la cual tiene una forma sinusoidal, tiene un pico amplio y sus lóbulos laterales son bajos y la ventana de Blackman-Harris la cual tiene un espectro resultante con un pico amplio utilizada para resolución en amplitud [10].

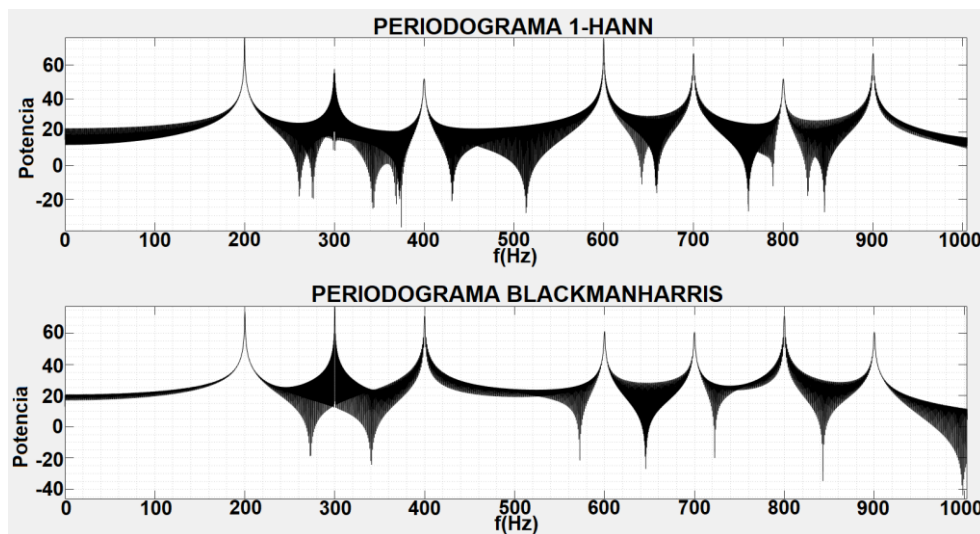


Figura 2.5: Periodograma con resolución en frecuencia y Periodograma con resolución en amplitud.

En la Figura 2.5 se puede presentar el desarrollo de la señal artificial multicomponentes diseñada al momento de realizar los 2 periodogramas para obtener la mayor cantidad de información tanto en frecuencia como en amplitud.

Una vez realizada esta acción, se procede a aplicar un doble filtro Savitsky-Golay. Este filtro sirve para suavizar los espectros, el cual es basado en la relación de una regresión lineal utilizando el método de mínimos cuadrados a través de un polinomio y el grado de este.

Este filtro se aplica tanto al espectro obtenido utilizando la ventana 1-Hann y Blackman-Harris. Una vez hecho esto, se suman los resultados obtenidos para tener un espectro que tenga las características tanto de resolución en frecuencia como resolución de amplitud.

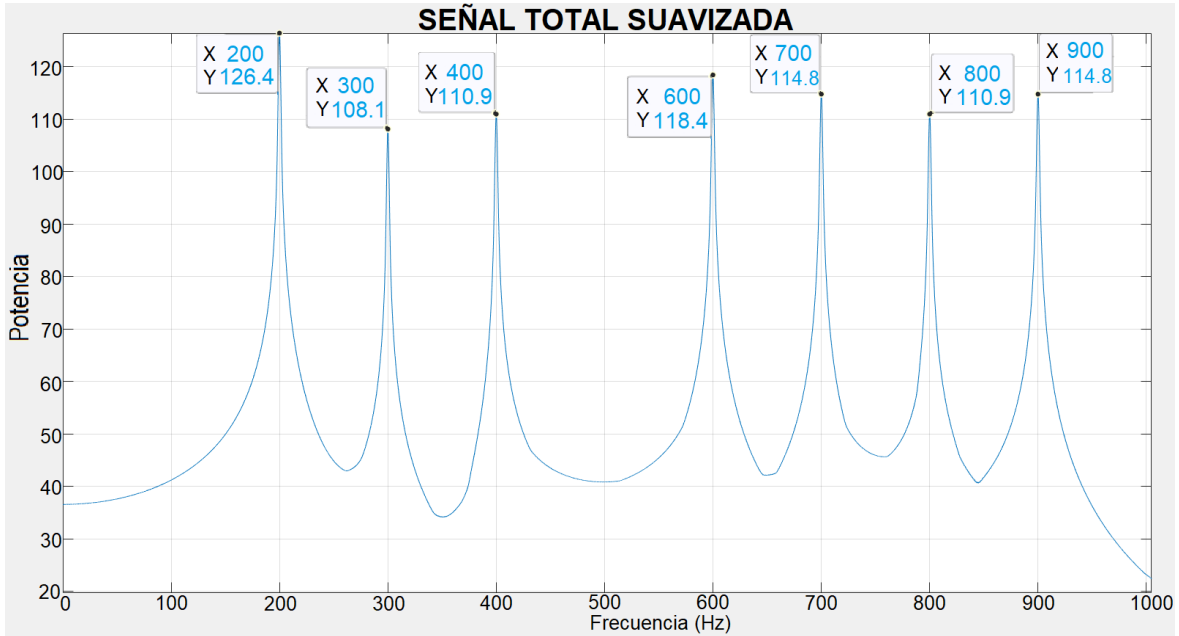


Figura 2.6: Periodograma total obtenido.

En la Figura 2.6 se muestran las 7 componentes de frecuencia correspondientes al escenario 1. Los lóbulos secundarios han sido desechados ya que no presentan información relevante respecto al comportamiento de la señal artificial utilizada.

Los nuevos datos pasan a ser analizados a través de ventanas deslizantes utilizando 1-Hann y Blackman-Harris, utilizando un traslape del 50%. Se define el tamaño de la ventana y se procede a encontrar el espectrograma usando la función *spectrogram* definida en el software de Matlab.

En la Figura 2.7, se muestra la conducta de las frecuencias de estudio en el dominio temporal, adicional a esto, muestra la característica en amplitud (potencia) usando la barra de calor.

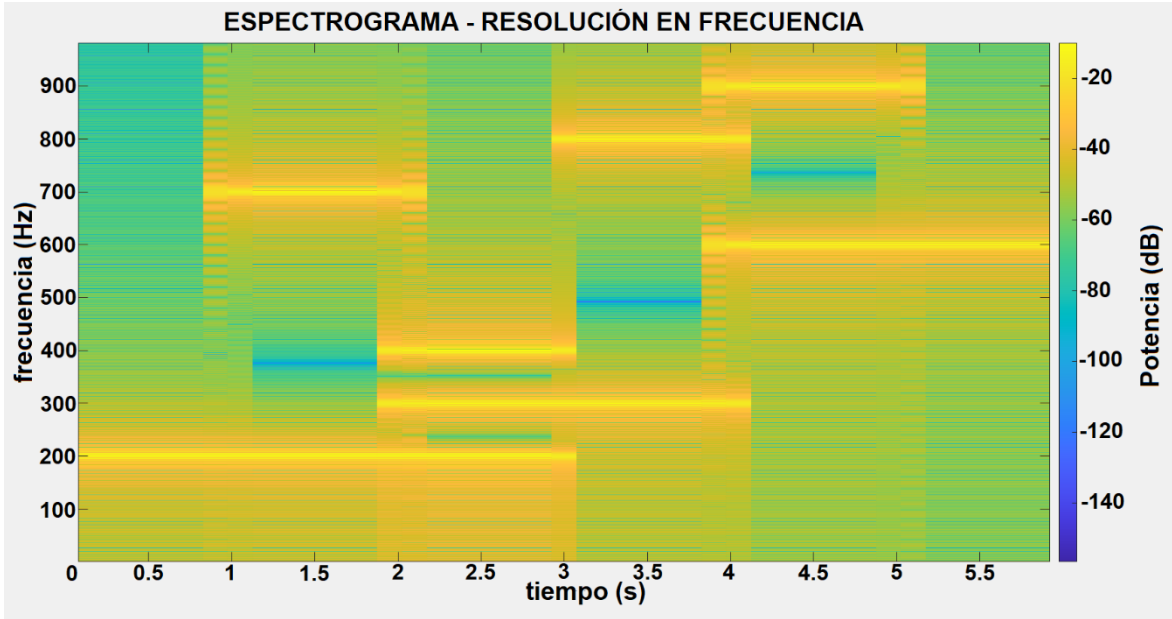


Figura 2.7: Espectrograma obtenido utilizando resolución en frecuencia.

El Análisis en Frecuencia da como resultado las componentes que se encuentran inmersas en la señal diseñada. Para complementar la información se necesita el análisis en tiempo de la señal de estudio.

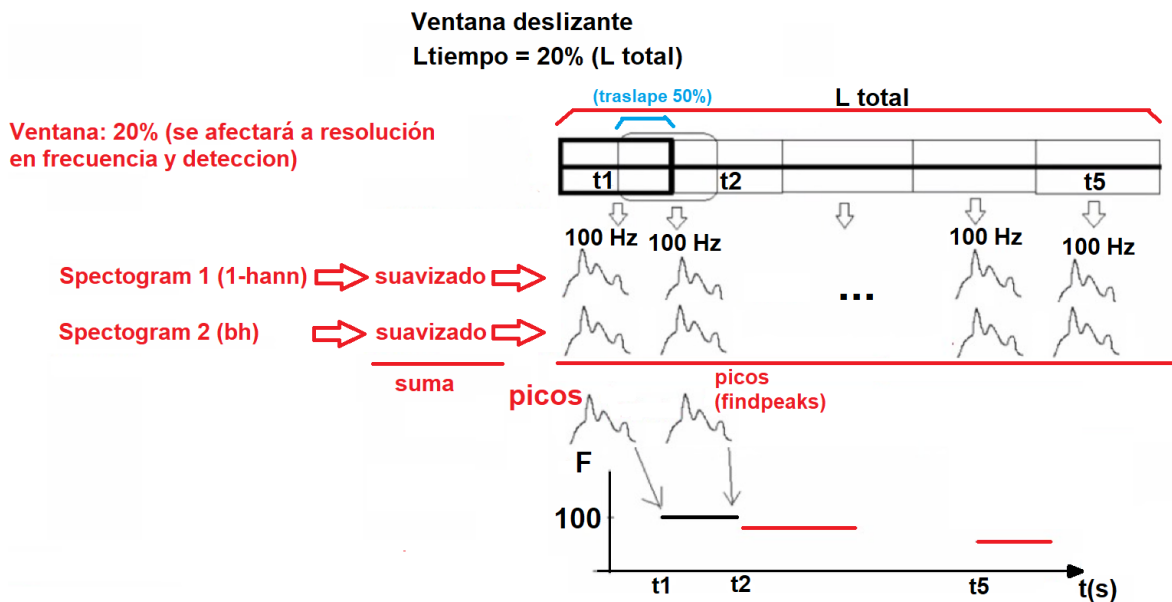


Figura 2.8: Diagrama comportamiento de la señal en tiempo.

La Figura 2.8 muestra el diagrama para obtener la información de frecuencia de cada una de las componentes previamente encontradas. Para ello se utiliza una ventana deslizante,

en cada instante se encontrará un espectro el cual tendrá determinadas componentes de frecuencia.

Este procedimiento se realiza mediante las ventanas 1-Hann y Blackmanharris debido a la resolución en frecuencia y tiempo respectivamente. Al realizar la suma de los espectros encontrados, se podrá obtener los instantes de tiempo en los cuales existe una componente específica en frecuencia.

En la Figura 2.9, se presenta el algoritmo con el cual se logran obtener los picos más representativos en cada uno de los espectros analizados. Dichos espectros son obtenidos con las diferentes ventanas deslizantes obtenidas.

Para iniciar este proceso se realiza un lazo que recorrerá el vector de frecuencias a estudiar, el cual está conformado por las frecuencias de mayor relevancia. Cada una de estas frecuencias será utilizada para comparar con las frecuencias obtenidas dentro de un bucle interno.

El bucle interno utiliza una variable auxiliar b la cual recorre por todos los espectros obtenidos en diferentes instantes de tiempo. Se obtiene el módulo de la potencia correspondiente al espectro b y se realiza la normalización. Dichos valores normalizados son utilizados en la función *findpeaks*, la cual entrega los valores picos de frecuencia en el espectro.

Estos picos encontrados son almacenados en un vector *frec*. Dicho vector debe ser recorrido a través de un lazo para posteriormente ser almacenado en una matriz de frecuencias llamada *mFrecuencia*.

Para poder almacenar las frecuencias identificadas, se lleva a cabo una comparación de intervalo. El objetivo de la comparación es poder definir cuál es la frecuencia a la que más se acerca. Como ejemplo, se tiene dos frecuencias de 500 [Hz] y 600 [Hz], la frecuencia obtenida a través del proceso mostrado es igual a 502 [Hz]. Con estos valores, es evidente que la frecuencia obtenida pertenece al comportamiento de la frecuencia de estudio de 500 [Hz].

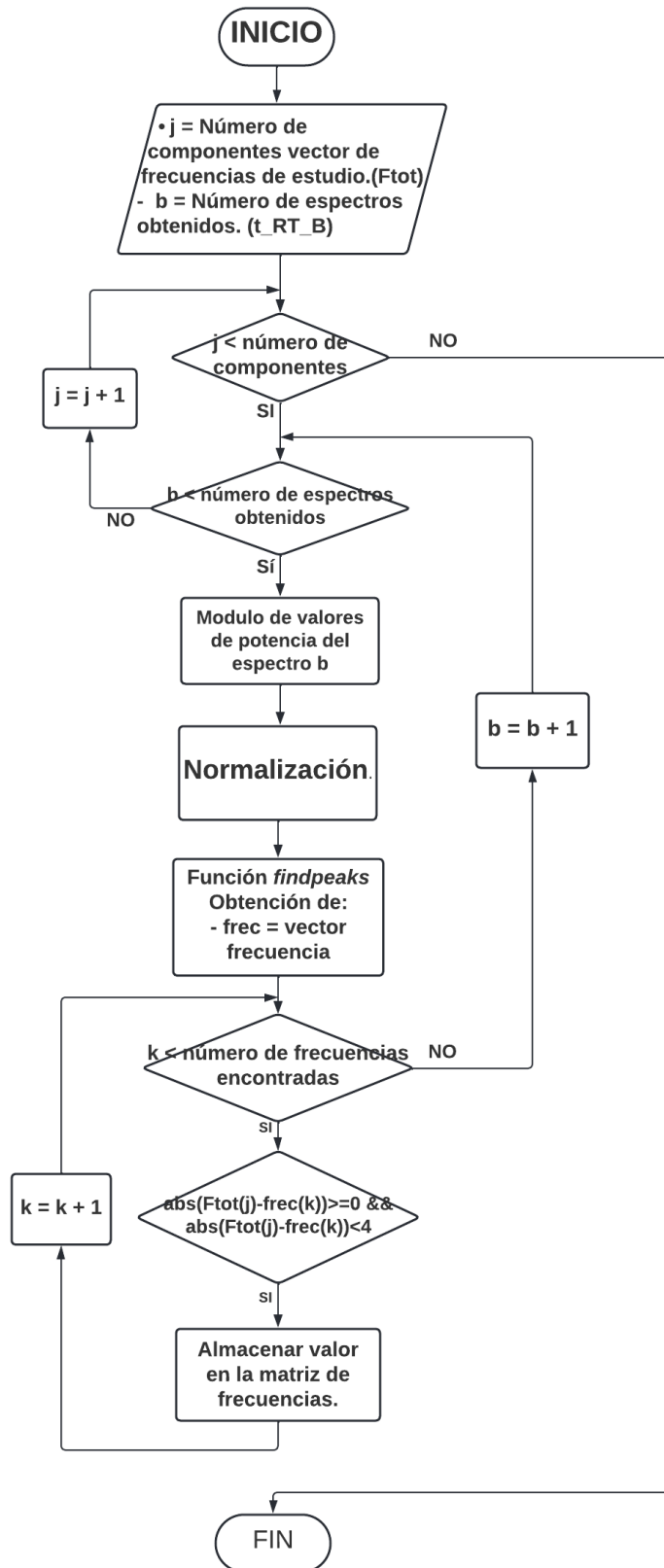


Figura 2.9: Diagrama de flujo detección y almacenamiento de frecuencias de interés.

Este algoritmo se repite hasta poder culminar con todo el vector de frecuencias a analizar, para posteriormente mostrar gráficamente el comportamiento de las frecuencias en el dominio temporal.

A continuación, se muestra el código descrito en el diagrama de flujo anterior.

```

%% 8. Find Peaks aplicado a cada uno de los espectros
(39)
disp('2.- Carga Computacional Metodo Inteligente')
tic
for j=1:length(Ftot) % Lazo: recorre vector de
frecuencias utilizadas.

    for b=1:length(t_RT_B) % Lazo: recorre todos los
espectros obtenidos.
        x = abs(P_RT_B(:,b)); % Modulo de los
complejos. P_RT_B
        x = x./max(x);

[power,frec]=findpeaks(x,f_RT_B,'MinPeakProminence',0.
1,'MinPeakWidth',0.1,'MinPeakDistance',4);

        for k=1:length(frec) % Lazo: almacenamiento de
picos encontrados en cada espectro
            if abs(Ftot(j)-frec(k))>=0 && abs(Ftot(j)-
frec(k))<4
                mFrecuencia(j,b)=frec(k);
                mAmp(j,b)=power(k);
            end
        end
    end
end
toc
for m=1:length(mFrecuencia(:,1)) % Lazo: encera los
valores del vector que no tienen picos.
    for n=1:length(mFrecuencia)
        if mFrecuencia(m,n)==1
            mFrecuencia(m,n)=0;
        end
    end
end
end

%% 9. Grafica Tiempo - Frecuencia Final

figure(9)

```



```

subplot(2,1,1);
for m=1:length(mFrecuencia(:,1)) % Lazo: grafica
continua de cada vector de frecuencias.
    hold on
    grid on
    plot(t_RT_B,mFrecuencia(m,:), '*','LineWidth',3);
    title('DIAGRAMA TIEMPO-FRECUENCIA METODO
INTELIGENTE');
    xlabel('Tiempo [s]')
    ylabel('Frecuencia [Hz]')
end
subplot(2,1,2);
hold on
plot (t1,[f1 f1], 'linewidth',2)
plot (t2,[f2 f2], 'linewidth',2)
plot (t3,[f3 f3], 'linewidth',2)
plot (t4,[f4 f4], 'linewidth',2)
plot (t5,[f5 f5], 'linewidth',2)
plot (t6,[f6 f6], 'linewidth',2)
plot (t7,[f7 f7], 'linewidth',2)
grid on;
title('DIAGRAMA TIEMPO-FRECUENCIA IDEAL')
xlabel('t(sec)')
ylabel('F(Hz)')

```

En la Figura 2.10 se muestra el diagrama Tiempo – Frecuencia obtenido a través del análisis de espectros. Cada uno de los espectros tiene un determinado número de componentes las cuales se muestran respecto al transcurso de tiempo.

La Figura 2.11 muestra una ampliación de los valores de tiempo de inicio y fin respecto a las componentes de 700 [Hz] y 300 [Hz].

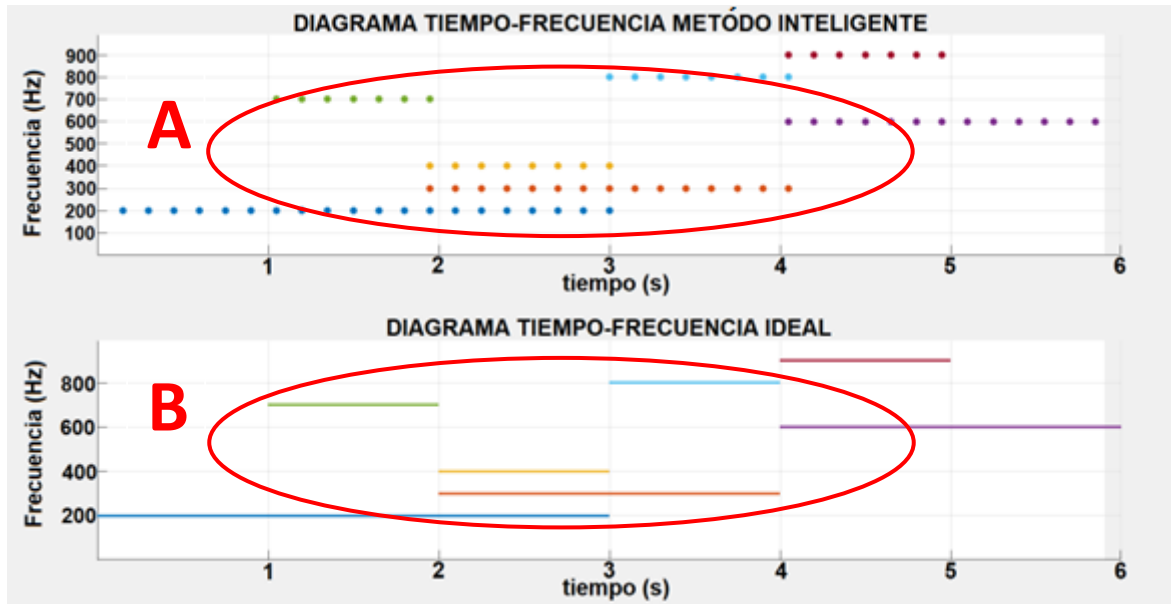


Figura 2.10: Diagrama Tiempo Frecuencia Obtenido vs Diagrama Tiempo Frecuencia Ideal

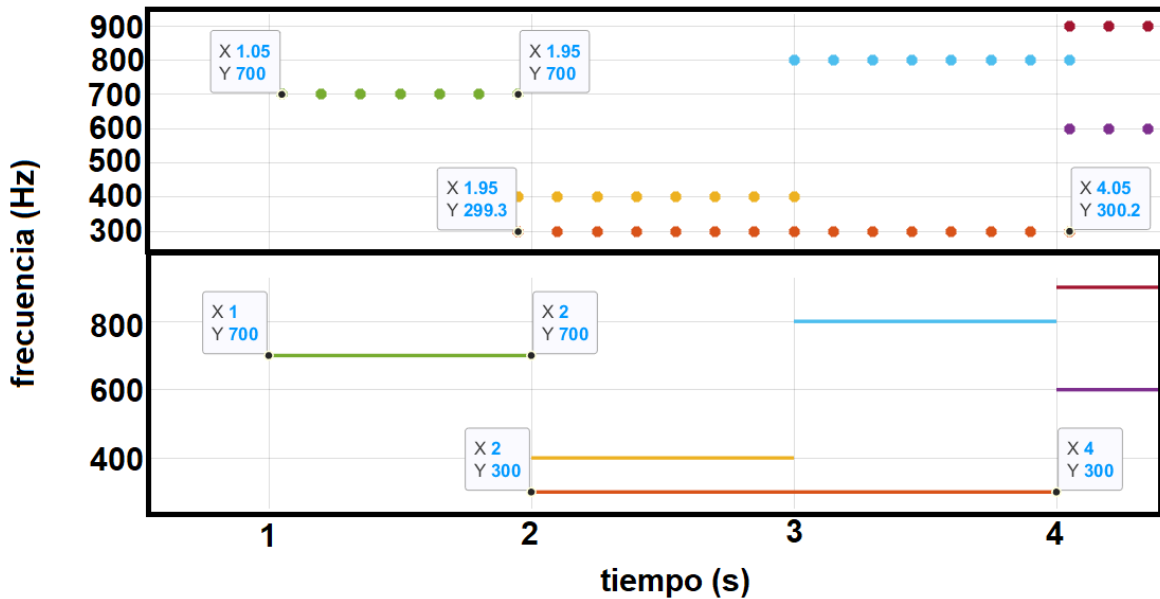


Figura 2.11: Acercamiento A y B respecto a la Figura 10

Para poder conocer la carga computacional, se utiliza una función establecida en el Software Matlab llamada *tic – toc* [14]. *Tic* inicia el cronómetro temporizador, mientras que *Toc* finaliza el cronómetro temporizador. Se registra el tiempo inicial y el tiempo final y se calcula la diferencia entre los 2, de esta manera se obtiene el tiempo de ejecución de determinadas secciones de código.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

Este capítulo está destinado para el análisis y discusión de los resultados obtenidos con las 2 pruebas realizadas utilizando métodos tradicionales y métodos inteligentes realizados teniendo en cuenta la resolución en frecuencia, para componentes muy juntas, resolución en amplitud, para componentes con amplitudes pequeñas, comparación de la carga computacional que corresponde a cada uno de ellos.

3.1 RESULTADOS

3.1.1 Prueba 1.

La prueba 1 consiste en realizar la medición de la carga computacional de la STFT con la señal artificial diseñada utilizando la función 'tic-toc' para el cálculo y visualización del diagrama TF para el método tradicional.

La Tabla 3.1 presenta los resultados obtenidos en los 6 escenarios establecidos para este trabajo de titulación. Cada uno de los escenarios presentará la carga computacional obtenida y el diagrama Tiempo – Frecuencia encontrado.

Escenario 1: Los valores de amplitud son igual a uno, la separación entre las componentes de frecuencia es considerable y el tiempo de duración de cada una de las componentes previamente definido.

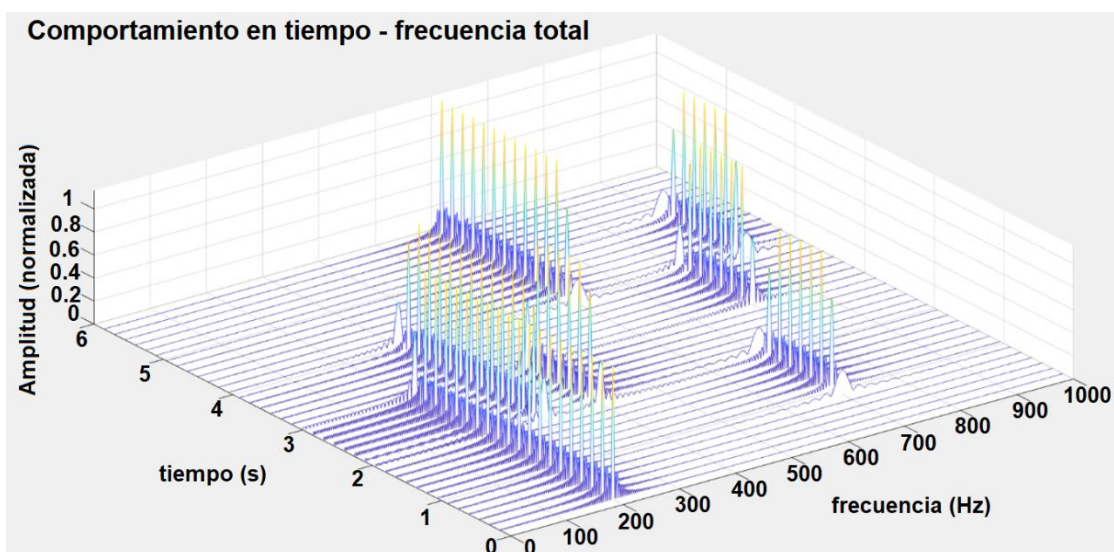


Figura 3.1: Gráfica tridimensional de los espectros obtenidos totales en el dominio del tiempo – frecuencia – amplitud obtenida en el Escenario 1.

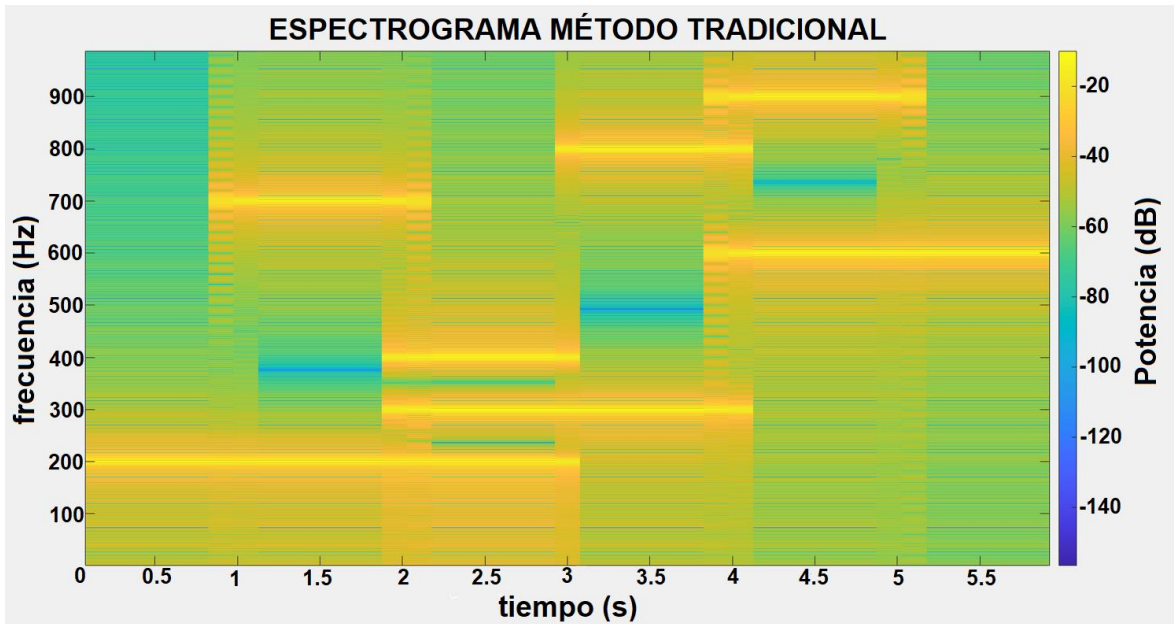


Figura 3.2: Espectrograma obtenido tomando en consideración las condiciones correspondientes al escenario 1.

Escenario 2: En este escenario se pone a prueba la detección de frecuencia ya que se establecen dos pares de componentes más cercanas una de otra.

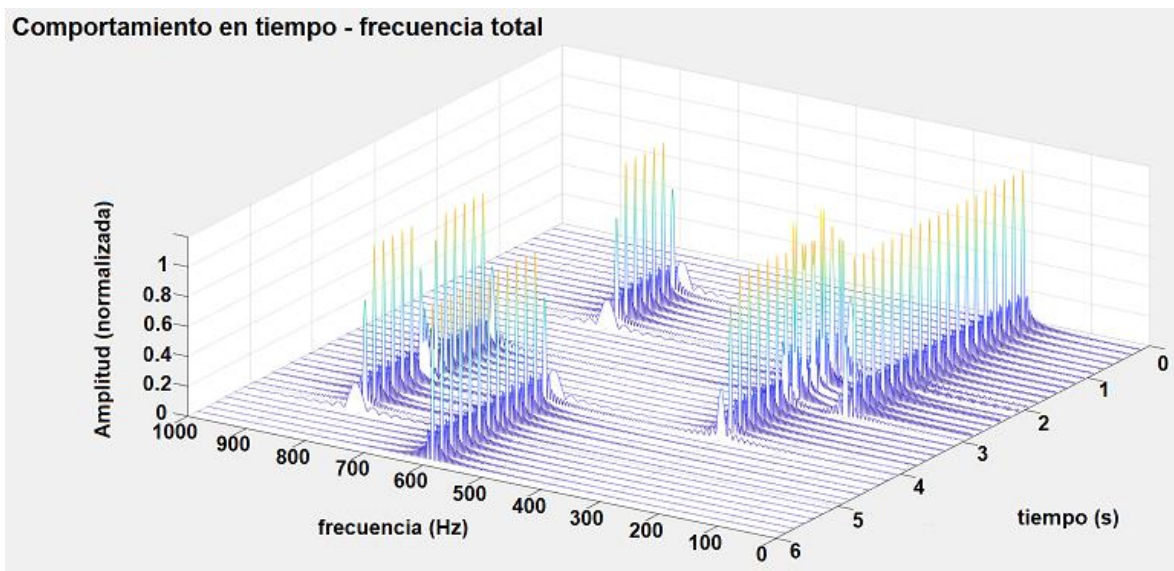


Figura 3.3: Gráfica tridimensional de los espectros obtenidos totales en el dominio del tiempo – frecuencia – amplitud obtenida en el Escenario 2.

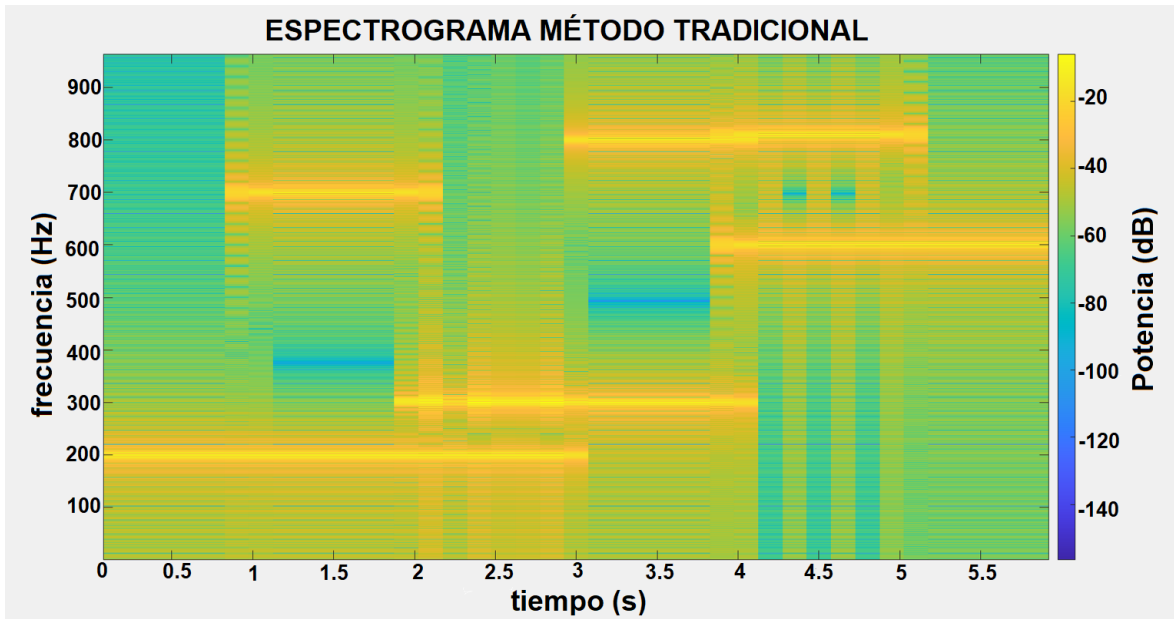


Figura 3.4: Espectrograma obtenido tomando en consideración las condiciones correspondientes al escenario 2.

Escenario 3: La característica de este escenario es la disminución de amplitudes de las diferentes componentes de frecuencia a analizar. Su reducción va del 50% al 3%, poniendo a prueba la detección en amplitud.

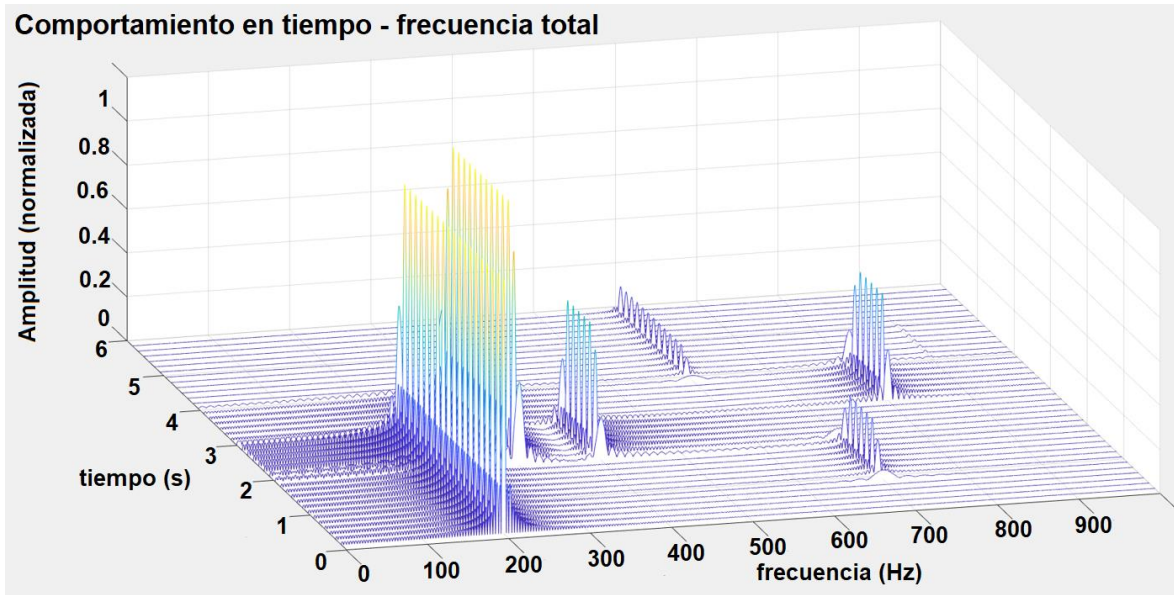


Figura 3.5: Gráfica tridimensional de los espectros obtenidos totales en el dominio del tiempo – frecuencia – amplitud obtenida en el Escenario 3.

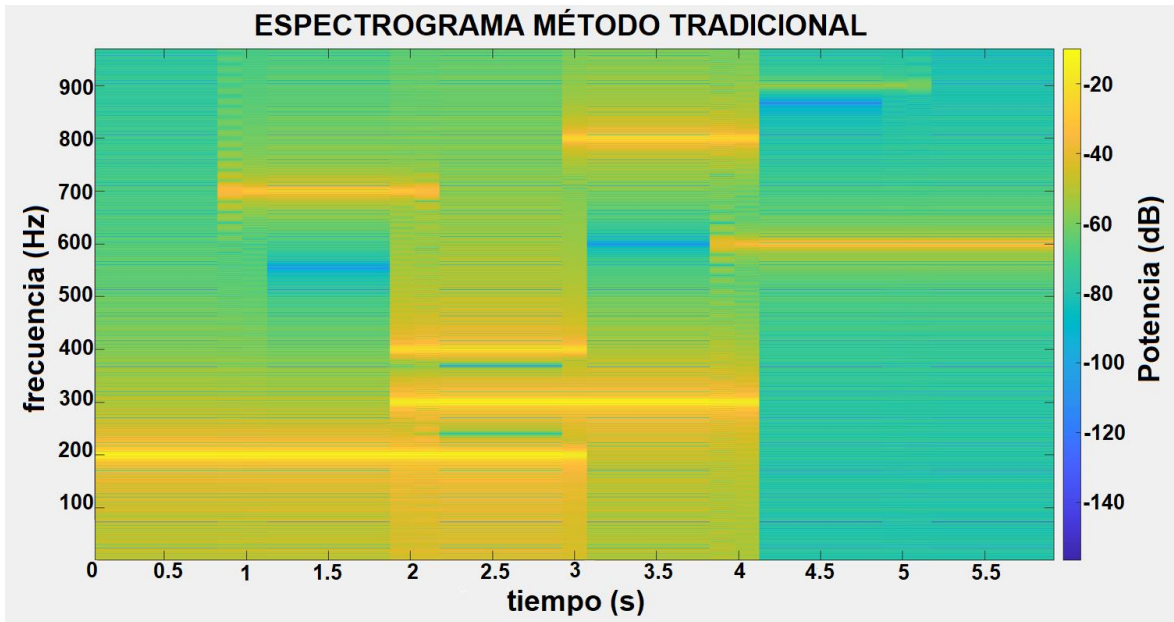


Figura 3.6: Espectrograma obtenido tomando en consideración las condiciones correspondientes al escenario 3.

Escenario 4: El objetivo es analizar el comportamiento de las componentes destacadas tanto con resolución en frecuencia como en amplitud.

Comportamiento en tiempo - frecuencia total

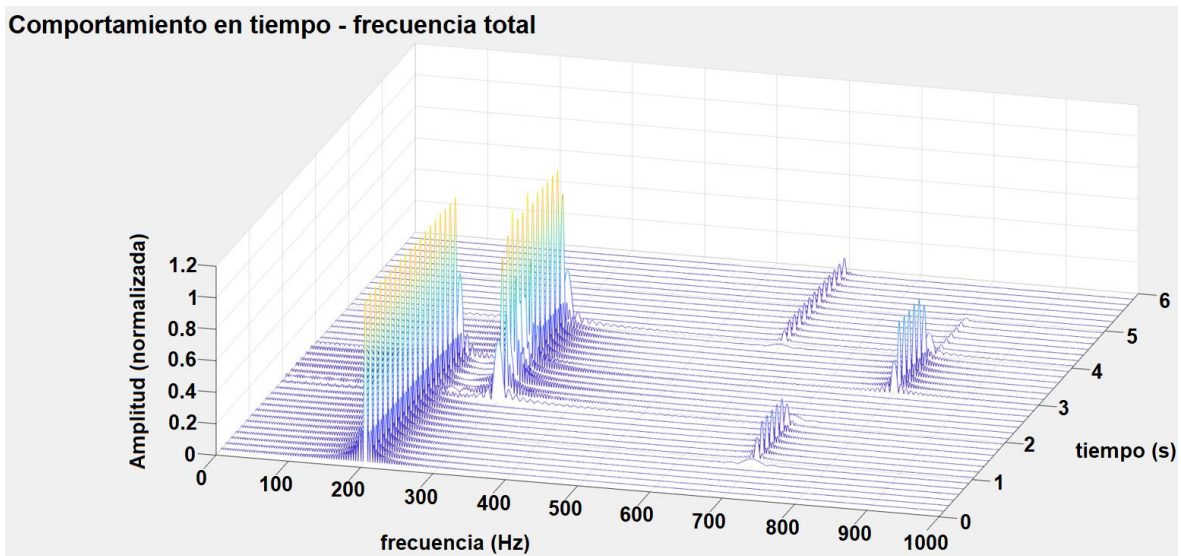


Figura 3.7: Gráfica tridimensional de los espectros obtenidos totales en el dominio del tiempo – frecuencia – amplitud obtenida en el Escenario 4.

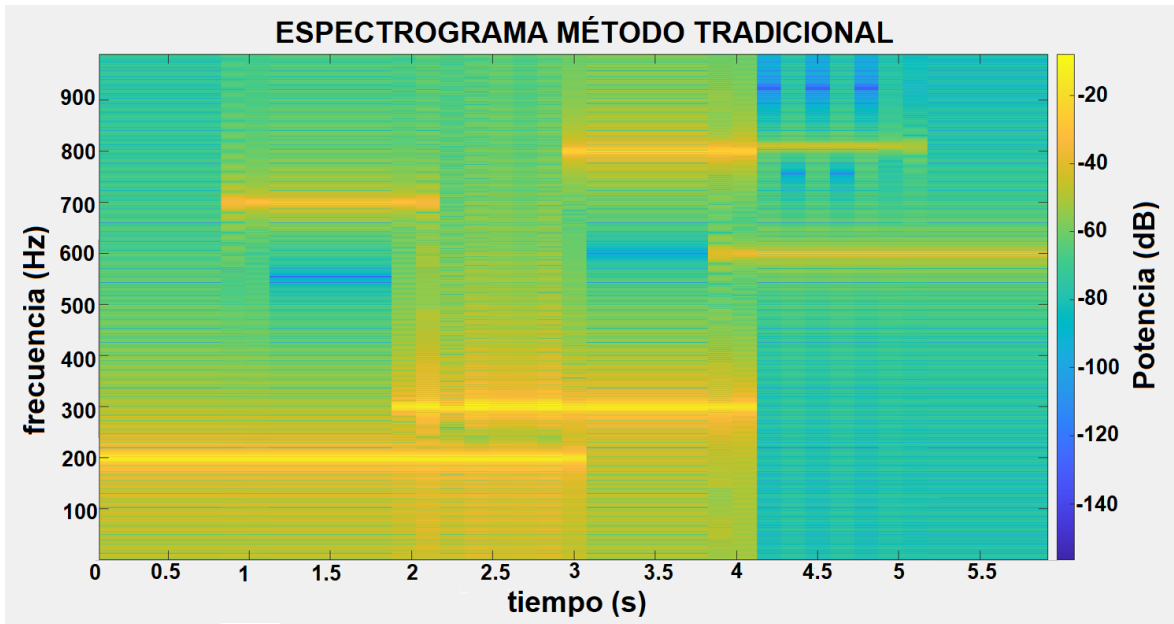


Figura 3.8: Espectrograma obtenido tomando en consideración las condiciones correspondientes al escenario 4.

Escenario 5: En este escenario está destinado para el uso de resolución en tiempo y analizar las componentes de interés, descartando las componentes que no aportan información relevante.

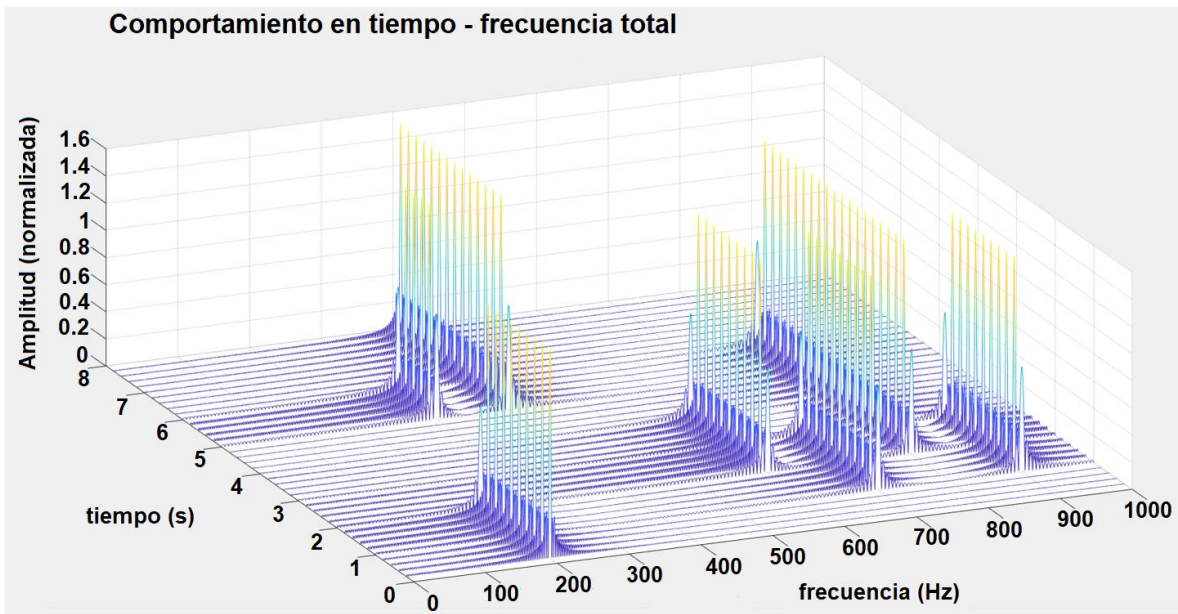


Figura 3.9: Gráfica tridimensional de los espectros obtenidos totales en el dominio del tiempo – frecuencia – amplitud obtenida en el Escenario 5.

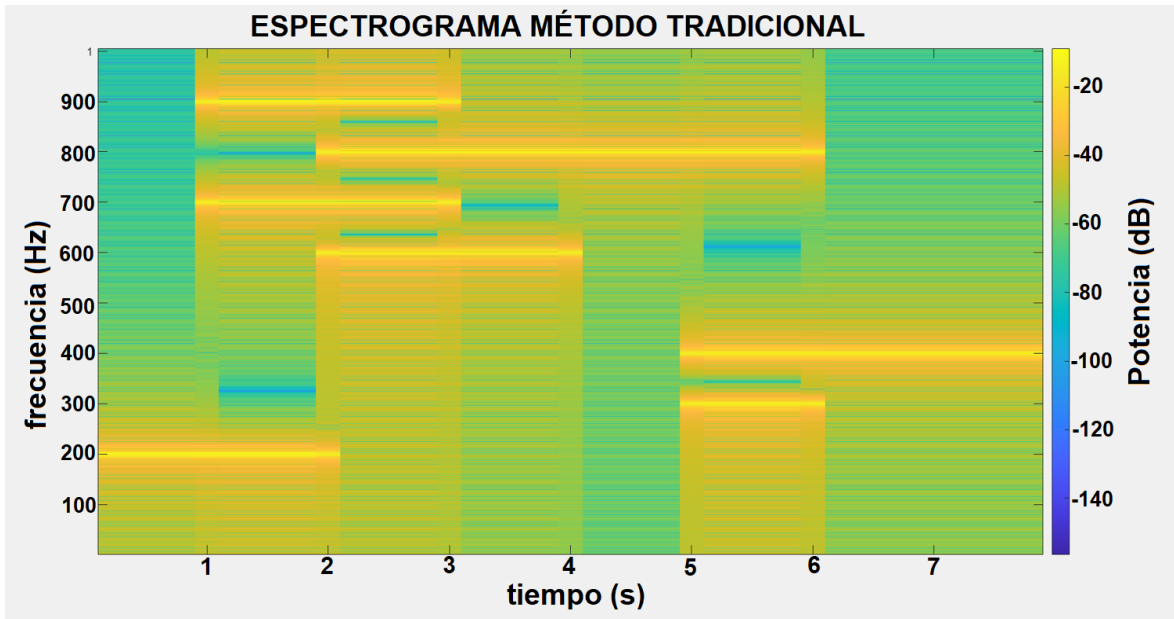


Figura 3.10: Espectrograma obtenido tomando en consideración las condiciones correspondientes al escenario 5.

Escenario 6: Se analiza el comportamiento de las componentes de interés al agregarles ruido blanco gaussiano (AGWN) con un valor de 25 dB.

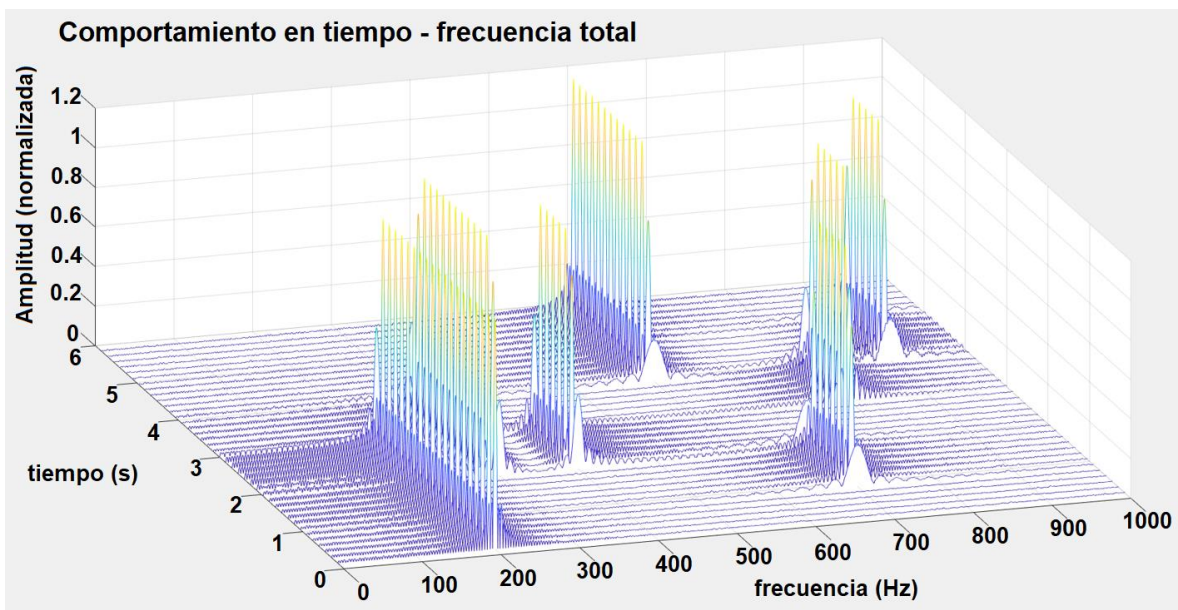


Figura 3.11: Gráfica tridimensional de los espectros obtenidos totales en el dominio del tiempo – frecuencia – amplitud obtenida en el Escenario 6.

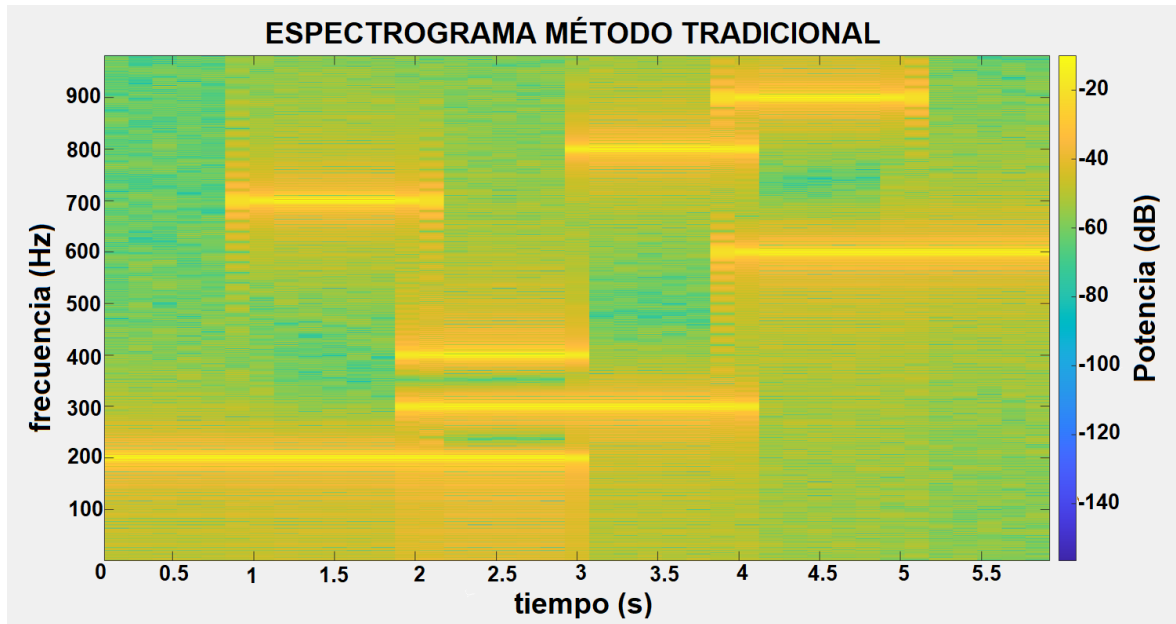


Figura 3.12: Espectrograma obtenido tomando en consideración las condiciones correspondientes al escenario 6.

A continuación, la Tabla 3.1 muestra los tiempos obtenidos respecto a la carga computacional utilizando métodos tradicionales. El tiempo de ejecución es relativamente muy bajo, pero, el tiempo de muestra de la gráfica es muy grande a tal punto de representar el 98 – 99% del valor total de Carga Computacional.

Tabla 3.1: Carga computacional usando Método Tradicional.

Escenario	Tiempo Ejecución [s]	Tiempo Gráfica [s]	Carga Computacional Total [s]
1. General	2.36	150.31	152.67
2. Resolución en frecuencia	2.26	153.77	156.03
3. Resolución en amplitud	2.19	150.31	152.5
4. Resolución en frecuencia y amplitud	2.17	191.48	193.65
5. Resolución en tiempo	2.17	220.51	222.68

6. General con ruido AWGN - 25 [dB]	2.22	208.53	210.75
--	------	--------	--------

3.1.2 Prueba 2.

La prueba 2 consiste en realizar la medición de la carga del nuevo algoritmo con la señal artificial diseñada utilizando la función 'tic-toc' para el cálculo y visualización del diagrama TF.

En la Tabla 3.2 se presentan los resultados obtenidos en los 6 escenarios establecidos para este trabajo de titulación. Cada uno de los escenarios presentará la carga computacional obtenida y el diagrama Tiempo – Frecuencia encontrado.

Escenario 1: Los valores de amplitud son igual a uno, la separación entre las componentes de frecuencia es considerable y el tiempo de duración de cada una de las componentes previamente definido.

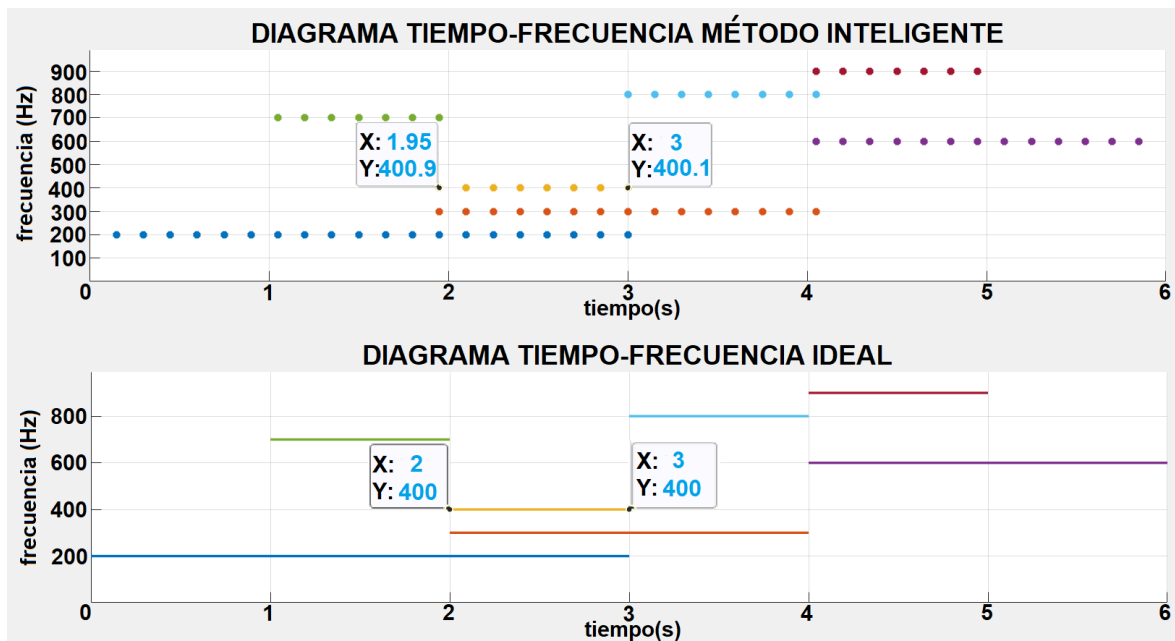


Figura 3.13: Comparación del diagrama Tiempo-Frecuencia correspondiente al escenario 1.

Escenario 2: En este escenario se pone a prueba la detección de frecuencia ya que se establecen dos pares de componentes más cercanas una de otra.

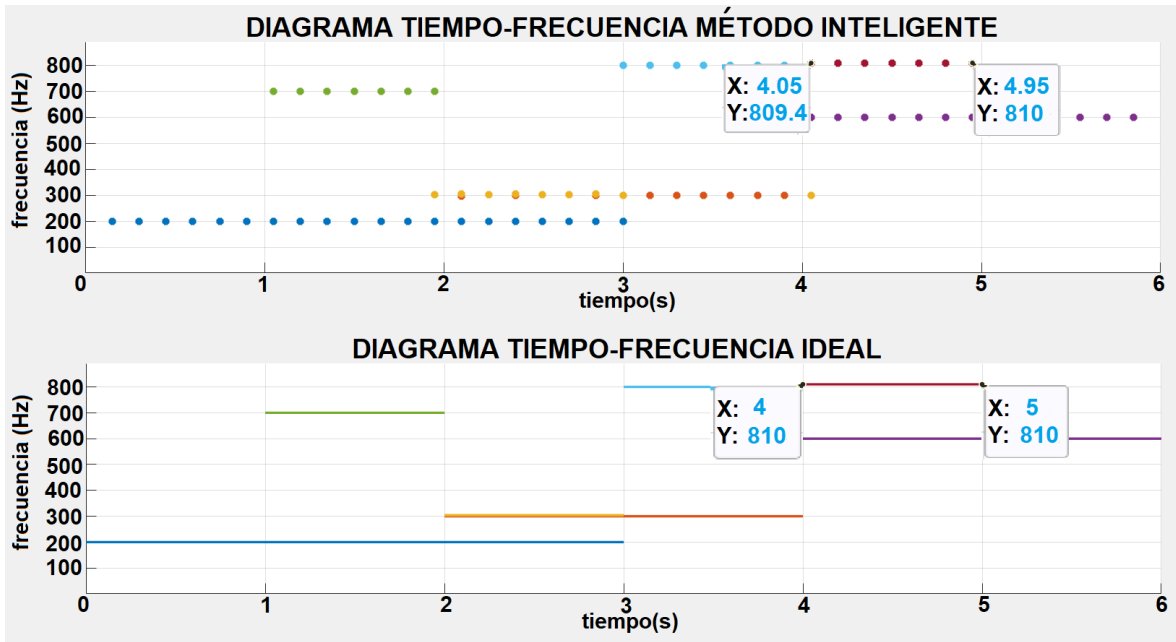


Figura 3.14: Comparación del diagrama Tiempo-Frecuencia correspondiente al escenario 2.

Escenario 3: La característica de este escenario es la disminución de amplitudes de las diferentes componentes de frecuencia a analizar. Su reducción va del 50% al 3%, poniendo a prueba la detección en amplitud.

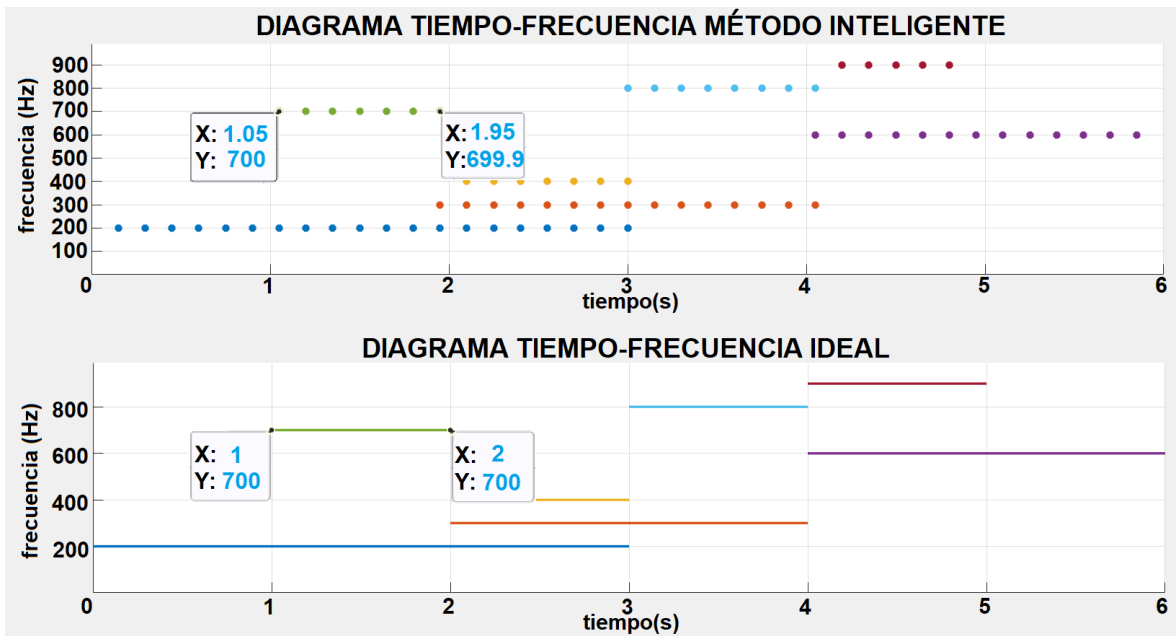


Figura 3.15: Comparación del diagrama Tiempo-Frecuencia correspondiente al escenario 3.

Escenario 4: El objetivo es analizar el desarrollo de las componentes destacadas tanto con resolución en frecuencia como en amplitud.

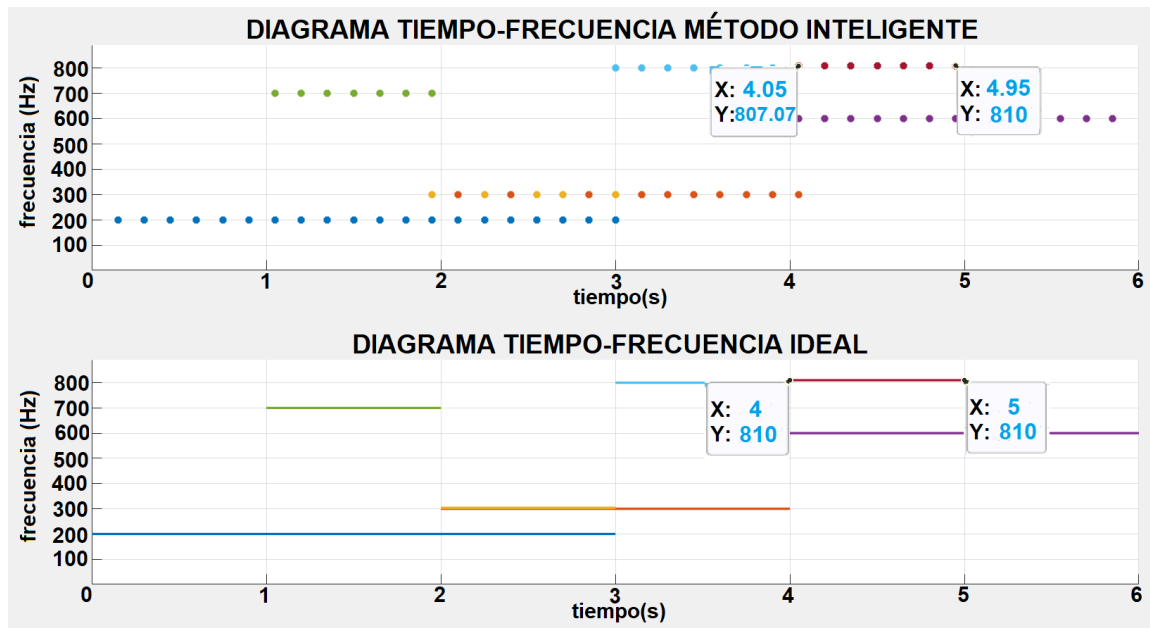


Figura 3.16: Comparación del diagrama Tiempo-Frecuencia correspondiente al escenario 4.

Escenario 5: En este escenario está destinado para el uso de resolución en tiempo y analizar las componentes de interés, descartando las componentes que no aportan información relevante.

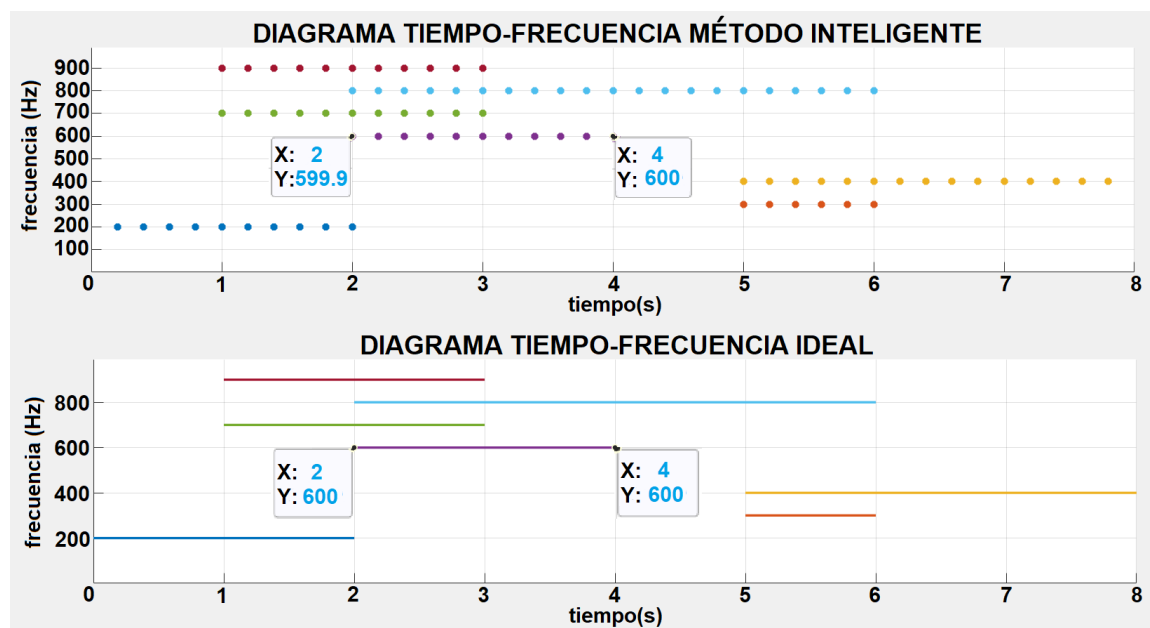


Figura 3.17: Comparación del diagrama Tiempo-Frecuencia correspondiente al escenario 5.

Escenario 6: Se analiza el comportamiento de las componentes de interés al agregarles ruido blanco gaussiano (AGWN) con un valor de 25 dB.

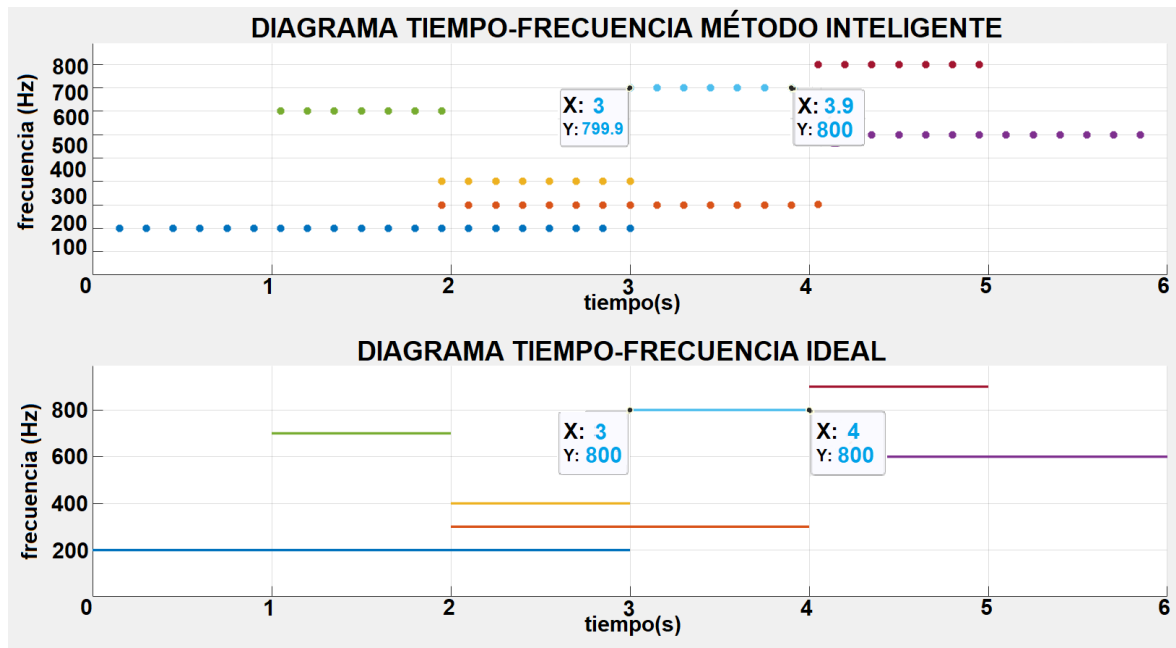


Figura 3.18: Comparación del diagrama Tiempo-Frecuencia correspondiente al escenario 6.

Tabla 3.2: Carga computacional usando Métodos Inteligentes

Escenarios	Tiempo Ejecución [s]	Tiempo Gráfica [s]	Carga Computacional Total [s]
1. General	8.24	5.84	14.08
2. Resolución en frecuencia	8.26	4.26	12.52
3. Resolución en amplitud	8.15	6.08	14.23
4. Resolución en frecuencia y amplitud	8.12	6.11	14.23
5. Resolución en tiempo	10.62	8.09	18.71
6. General con ruido AWGN - 25 [dB]	8.61	6.30	14.91

3.1.3 COMPARACIÓN RESULTADOS

Al obtener los resultados de carga computacional, se observa la reducción de la carga computacional en cada uno de los escenarios planteados. Para obtener estos resultados, se realizó la simulación en el equipo descrito al inicio del documento.

Escenario 1: Este escenario se caracterizaba por tener los valores de amplitud igual a 1, frecuencias separadas y lapsos establecidos. Utilizando métodos tradicionales se obtuvo una carga computacional de aproximadamente 153 segundos (2 minutos 33 segundos), mientras que, usando el nuevo algoritmo desarrollado, la carga computacional es de 14 segundos aproximadamente. Teniendo una reducción de 9 veces el valor de la carga computacional usando el nuevo algoritmo.

Escenario 2: La característica de este escenario consistía en acercar 2 pares de frecuencias, manteniendo el valor de amplitud y el lapso. La carga computacional de la primera prueba fue de aproximadamente 156 segundos (2 minutos 36 segundos) mientras que, en la segunda prueba, la carga computacional fue de 12 segundos aproximadamente. Esto nos indica la reducción de aproximadamente 92% en comparación a la carga computacional de la primera prueba.

Escenario 3: Para el análisis se realiza variaciones en la amplitud de cada una de las componentes de la señal artificial, manteniendo las características generales de lapso y separación entre frecuencias. La carga computacional obtenida en la prueba 1 es de aproximadamente 153 segundos (2 minutos 33 segundos). La carga computacional en la prueba 2 es de aproximadamente 14 segundos, teniendo una reducción del 91% aproximadamente respecto a la primera prueba.

Escenario 4: Este escenario se compone por las variaciones en amplitud y la aproximación entre 2 pares de frecuencias teniendo un aumento en la carga computacional. Este valor obtenido es de aproximadamente 194 segundos (3 minutos 14 segundos) mientras que la carga computacional de la segunda prueba es de 14 segundos aproximadamente.

Escenario 5: La variación que se realiza es cambiar los valores de tiempo y aumentar la duración total de la señal, poniendo a prueba la capacidad de los algoritmos utilizados. Usando métodos tradicionales, la carga computacional obtenida es de 222.68 segundos (3 minutos 42 segundos) y la carga computacional obtenida usando el nuevo algoritmo es de aproximadamente 19 segundos.

Escenario 6: En el último escenario se utilizan los parámetros del escenario 1 aumentando ruido blanco gaussiano de 25 dB. La carga computacional obtenida en la primera prueba

es de aproximadamente 211 segundos (3 minutos 31 segundos) y en la segunda prueba se obtuvo una carga computacional de aproximadamente 15 segundos, teniendo una reducción de aproximadamente 93% respecto a la carga de métodos tradicionales.

Tabla 3.3: Comparación de carga computacional usando métodos tradicionales vs métodos inteligentes.

Escenarios	Carga Computacional Método Tradicional [s]	Carga Computacional Métodos Inteligentes [s]	% de Reducción de Carga Computacional
1. General	152.67	14.08	90.777
2. Resolución en frecuencia	156.03	12.52	91.976
3. Resolución en amplitud	152.5	14.23	90.669
4. Resolución en frecuencia y amplitud	193.65	14.23	92.652
5. Resolución en tiempo	222.68	18.71	91.598
6. General con ruido AWGN - 25 [dB]	210.75	14.91	92.925

En la Tabla 3.3, se puede observar una reducción del 91% \pm 1% respecto a la carga computacional obtenida por métodos tradicionales. Adicional a esto, se obtiene un error mínimo respecto a los valores en frecuencia obtenido en cada uno de los escenarios analizados. Esta comparación se la realizó tomando pares de valores obtenidos con el método inteligente desarrollado y los valores ideales utilizados para el diseño de la señal compuesta.

Tabla 3.4: Error porcentual obtenido en tiempo y frecuencia.

	IDEAL		METODO DESARROLLADO		% de Error en Frecuencia	% de Error en Tiempo
	Tiempo [s]	Frecuencia [Hz]	Tiempo [s]	Frecuencia [Hz]		
Escenario 1	2	400	1.95	400.9	0.22	2.56
	3	400	3	400.1	0.03	0.00

Escenario 2	4	810	4.05	809.4	0.07	1.23
	5	810	4.95	810	0.00	1.01
Escenario 3	1	700	1.05	700	0.00	4.76
	2	700	1.95	699.9	0.01	2.56
Escenario 4	4	810	4.05	807.07	0.36	1.23
	5	810	4.95	810	0.00	1.01
Escenario 5	2	600	2	599.9	0.02	0.00
	4	600	4	600	0.00	0.00
Escenario 6	3	800	3	799.9	0.01	0.00
	4	800	3.9	800	0.00	2.56

En la Tabla 3.4 se muestra el error obtenido al utilizar el nuevo algoritmo desarrollado. En frecuencia, se tiene un error de menos del 1% respecto a los valores ideales utilizados para la comparación. Cada uno de los escenarios tiene características diferentes por lo que el porcentaje de error varía entre ellos.

El error obtenido en el análisis de tiempo es un poco más notable. Se toma los valores de inicio y fin de una componente en específico y se compara con los obtenidos, teniendo un error de menos del 5% en el escenario 3, en el resto de los escenarios se tiene un error menor al 3%.

3.2 CONCLUSIONES

Al realizar la Prueba 1, reflejó que la carga computacional obtenida en cada uno de los escenarios aumenta dependiendo de la característica del escenario planteado. A partir del escenario 4, aumenta la carga computacional un 30% en comparación con los tres primeros escenarios estudiados.

La Prueba 2, mostró el comportamiento y carga computacional del algoritmo desarrollado, teniendo como resultado una disminución considerable respecto a la carga computacional obtenida en la Prueba 1. El promedio de reducción de carga computacional en comparación con la Prueba 1 es del 90%.

El error obtenido en el dominio de tiempo y frecuencia reflejó que al usar métodos inteligentes se puede utilizar la novena parte del tiempo utilizado con los métodos tradicionales, a cambio de tener un error mínimo del 1% en el dominio de la frecuencia y entre un 3% a un 5% en el dominio temporal.

Los métodos tradicionales requieren una mayor carga computacional ya que realizan el análisis de todo el espectro, generando la imagen para todos los valores que conforman dicho espectro, mientras que el método inteligente se especializa en analizar valores críticos y de interés, disminuyendo así la carga computacional.

Si bien los resultados obtenidos mediante las dos pruebas planteadas con sus seis escenarios respectivamente fueron favorables ya que reflejan la disminución en la carga computacional y se tiene mínimos porcentajes de error respecto a la resolución en frecuencia, el porcentaje de error en resolución en tiempo es apreciable en componentes determinadas, ya sea por el aumento o disminución de los límites de las componentes.

3.3 RECOMENDACIONES

Este estudio muestra un análisis general entre el método tradicional basado en la STFT contra el algoritmo desarrollado usando Métodos Inteligentes en el dominio tiempo – frecuencia, se recomienda ahondar y desarrollar nuevas pruebas con escenarios alternos, como aumentar el valor de ruido AWGN, adicionar nuevas componentes a la señal de estudio, utilizar ventanas deslizantes diferentes a 1-Hann y Blackman Harris como Kaiser-Bessel, y utilizando otros tipos de métodos tradicionales como la Sincronizada de Fourier o la Reasignada de Fourier.

Cada uno de los escenarios implementados tienen un valor diferente de carga computacional, dependiendo de la prueba a la cual fue sometido usando un equipo con especificaciones descritas en la parte superior, se recomienda verificar los requerimientos del equipo en el cual de vaya a realizar nuevas pruebas ya que los datos obtenidos pueden cambiar como aumentar la carga computacional, al usar un equipo con un procesador con menor número de núcleos y generación, o disminuir la carga computacional al usar un equipo con mejores características del microprocesador.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] S. L. Marple, "Time-frequency signal analysis: Issues and alternative methods", *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, pp. 329-332, 1998, DOI: 10.1109/TFSA.1998.721427.

- [2] M. R. Canal, "Comparison of wavelet and short time Fourier Transform methods in the analysis of EMG signals", *J Med Syst*, vol. 34, n.º 1, pp. 91-94, feb. 2010, DOI: 10.1007/S10916-008-9219-8/METRICS.
- [3] X. Li *et al.*, "Comparison of different time-frequency analysis methods for sparse representation of PD-induced UHF signal", *China International Conference on Electricity Distribution, CICED*, vol. 2016-September, sep. 2016, DOI: 10.1109/CICED.2016.7575902.
- [4] Q. Zhu, Y. Wang, y G. Shen, "Research and comparison of time-frequency techniques for nonstationary signals", *J Comput (Taipei)*, vol. 7, n.º 4, pp. 954-958, 2012, DOI: 10.4304/JCP.7.4.954-958.
- [5] R. Alvarez, E. Borbor, y F. Grijalva, "Comparison of methods for signal analysis in the time-frequency domain", *2019 IEEE 4th Ecuador Technical Chapters Meeting, ETCM 2019*, nov. 2019, DOI: 10.1109/ETCM48019.2019.9014860.
- [6] A. M. Moros *Caracterización de señales de precipitación mediante la transformada de Fourier y transformada Wavelet. [online]. Disponible en: <http://hdl.handle.net/10554/6289>.*
- [7] A. M. M. Vivas, *CARACTERIZACIÓN DE SEÑALES DE PRECIPITACIÓN MEDIANTE LA TRANSFORMADA DE FOURIER Y TRANSFORMADA WAVELET*, Bogota: PONTIFICIA UNIVERSIDAD JAVERIANA, 2010.
- [8] M. Z. Castaño, *Análisis de Señales Mediante STFT y Wavelet. Aplicación a Defectología en Rodamiento*, Leganés: Universidad Carlos III de Madrid, 2010.
- [9] Celorio Guamán, B. L (2023). *Estudio y comparación de métodos modernos para la determinación del comportamiento en tiempo - frecuencia de una señal*. 49 páginas. Quito: EPN. *Disponible en: <http://bibdigital.epn.edu.ec/handle/15000/23824>*
- [10] "Comprender FFTs y Funciones Ventana". <https://www.ni.com/es-cr/shop/data-acquisition/measurement-fundamentals-main-page/analog-fundamentals/understanding-ffts-and-windowing.html> (accedido 16 de julio de 2023)
- [11] MathWorks, "MATLAB stft - Short-time Fourier transform - MathWorks," *MathWorks*, [En línea]. Available: <https://es.mathworks.com/help/signal/ref/stft.html>. [Último acceso: 28 06 2023].
- [12] MathWorks, "Spectrogram using short-time Fourier transform - MathWorks," *MathWorks*, [En línea]. Available:

- <https://es.mathworks.com/help/signal/ref/spectrogram.html>. [Último acceso: 28 jun 2023].
- [13] “Encontrar los máximos locales - MATLAB findpeaks - MathWorks España”. <https://es.mathworks.com/help/signal/ref/findpeaks.html> [Último acceso: 28 jun 2023].
- [14] MathWorks, “MATLAB tic - Iniciar el cronómetro temporizador - MathWorks,” [En línea]. Available: <https://es.mathworks.com/help/matlab/ref/tic.html>. [Último acceso: 15 06 2023].
- [15] “Filtrado de Savitzky-Golay - MATLAB sgolayfilt - MathWorks España”. <https://es.mathworks.com/help/signal/ref/sgolayfilt.html> [Último acceso: 15 06 2023].
- [16] N. B. Gallagher, “Savitzky-Golay Smoothing and Differentiation Filter,” EIGENVECTOR, research incorporated, Manson, WA.
- [17] Schafer, R. (2011). What Is a Savitzky-Golay Filter? [Lecture Notes]. IEEE Signal Processing Magazine, 28(4), 111–117. doi:10.1109/msp.2011.941097
- [18] I. R. V. Quintero, “Implementación de un modelo del comportamiento de los niveles de concentración de PM10 utilizando herramientas de aprendizaje de máquina en la ciudad de Santa Marta,” Universidad del Magdalena, Santa Marta D.T.C.H, 2017.
- [19] M. F. A. F. Eduardo P. Serrano, "Metodos Tiempo-Frecuencia Basados en la Transformada Wavelet," Revista de Matematica: Teoria y Aplicaciones Cimpa-UCR - Universidad de Costa Rica, San José, San Pedro, Costa Rica, 2012.
- [20] Plante, F., Meyer, G., & Ainsworth, W. A. (1998). Improvement of speech spectrogram accuracy by the method of reassignment. IEEE Transactions on Speech and Audio Processing, 6(3), 282–287. doi:10.1109/89.668821

5 ANEXOS

ANEXO I. Código utilizado.

ANEXO II. Guía de Uso.

ANEXO I

```
% ----- MAIN PRINCIPAL -----
% ----- %
% Metodos Inteligentes
% Autor: Jonathan Chavez Jinez
%
% GRAFICAS PARA LAS 4 PRUEBAS
% Grafica 1: OK
% Grafica 2: OK
% Grafica 3: 1-hann, blackmanharris (SIN SUAVIZAR)
% Grafica 4: SUAVIZAR Y LUEGO SUMAR. (EN SUBPLOTS) 1-
hann suavizado, bh
% suavizado, suma de los 2 suavizados. Se suaviza para
eliminar los lobulos
% secundarios con el filtro sgolay.
% (EXPERIMENTAR: 1.- Suma y despues findpick. 2.-
Findpeack independientes
% y despues sumar vector de picos.
% Grafica 5: Tiempo
% Grafica 6: Tiempo-Frecuencia a partir de la funcion
spectogram y findpeaks
%-----
%-----%
% Escenario 1
% No hay requerimientos de rf ni de deteccion
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900;
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1;
% 1-hann como blackmanharris deberian encontrar TODA
LA INFORMACION.
% Escenario 2
% Requerimiento de resolucion en frecuencia (acercamos
2 pares de
% componentes); no hay requerimiento de deteccion.
% f1=200; f2=300; f3=304; f4=600; f5=700; f6=800;
f7=820;
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1;
% Solo 1-hann encuentra TODA LA INFORMACION, bh
encontraria a f2 y f3 como
% una sola frecuencia, f6 y f7 se encontrarian como 1.
(ANALIZAR LOS
% VALORES DE CERCANIA 300-304 y 800-820)
```

```

% Escenario 3
% Requerimiento de deteccion (disminuir la amplitud de
un par de componentes)
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900;
% a1=1; a2=0.1; a3=0.01; a4=0.001; a5=0.0001;
a6=0.00001; a7=0.000001;
% Solo bk encuentra TODA LA INFORMACION, 1-hann no
encontraria las
% amplitudes mas peque;as (ANALIZAR LOS VALORES DE
AMPLITUDES PARA NO
% DETECTAR)

% Escenario 4
% Requerimiento tanto en deteccion como en resolucion
en frecuencia
% f1=200; f2=300; f3=304; f4=600; f5=700; f6=800;
f7=810;
% a1=1; a2=1; a3=0.5; a4=0.1; a5=0.2; a6=0.4; a7=0.03;
% Unimos la informacion CARACTERISTICA de ambas,
suavizar y sumar los valores
% Lo de 1-hann NO DEBERIA APARECER EN bh.

% Escenario 5
% Cambiar duracion de las componentes
% t1=[0 2]; t2=[5 6]; t3=[5 8]; t4=[2 4]; t5=[1 3];
t6=[2 6]; t7=[1 3];

% Escenario 6
% Adicionar ruido AWGN de unos 25dB
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900;
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1;

%%
clc, close all, clear all

%% 1. Señal Multicomponente:
% Los datos deben irse descomentando dependiendo del
escenario a estudiar:

% 1.1 Frecuencias de componentes, ordenadas
ascendentemente:
f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 1

```

```

% f1=200; f2=300; f3=304; f4=600; f5=700; f6=800;
f7=810; % Escenario 2
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 3
% f1=200; f2=300; f3=304; f4=600; f5=700; f6=800;
f7=810; % Escenario 4
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 5
% f1=200; f2=300; f3=400; f4=600; f5=700; f6=800;
f7=900; % Escenario 6
Ftot=[f1 f2 f3 f4 f5 f6 f7]; % Vector de frecuencias
fmax=max(Ftot);

% % 1.2 Amplitudes:
a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; % Escenario
1
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; %
Escenario 2
% a1=1; a2=0.8; a3=0.5; a4=0.1; a5=0.2; a6=0.4;
a7=0.01; % Escenario 3
% a1=1; a2=1; a3=0.5; a4=0.1; a5=0.2; a6=0.4; a7=0.03;
% Escenario 4
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; %
Escenario 5
% a1=1; a2=1; a3=1; a4=1; a5=1; a6=1; a7=1; %
Escenario 6
Atot=[a1 a2 a3 a4 a5 a6 a7]; % Vector de amplitudes

% 1.3 Lapso de cada componente:
t1=[0 3]; t2=[2 4]; t3=[2 3]; t4=[4 6]; t5=[1 2];
t6=[3 4]; t7=[4 5]; % Escenario 1, 2, 3, 4, 6
% t1=[0 2]; t2=[5 6]; t3=[5 8]; t4=[2 4]; t5=[1 3];
t6=[2 6]; t7=[1 3]; % Escenario 5

tT = [t1,t2,t3,t4,t5,t6,t7];

%% 2. Creación de señal compuesta
Fs=10*fmax; % FSmin >= 2*Fmax Nyquist
Ts=1/Fs;
duracion = max(tT); % duración total componentes

t=0:Ts:duracion;

% Generacion de componentes por intervalos
comp1=a1*sin(2*pi*f1*t);
comp2=a2*sin(2*pi*f2*t);

```

```

comp3=a3*sin(2*pi*f3*t);
comp4=a4*sin(2*pi*f4*t);
comp5=a5*sin(2*pi*f5*t);
comp6=a6*sin(2*pi*f6*t);
comp7=a7*sin(2*pi*f7*t);

% ----- Escenarios 1,2,3,4,6 -----
-%
comp1=(t<=3).*comp1;
comp2=((t>=2)&(t<=4)).*comp2;
comp3=((t>2) & (t<=3)).*comp3;
comp4=((t>=4)&(t<=6)).*comp4;
comp5=((t>=1)&(t<=2)).*comp5;
comp6=((t>=3)&(t<=4)).*comp6;
comp7=((t>=4)&(t<=5)).*comp7;
% -----%

% ----- Escenario 5 -----%
% comp1=(t<=2).*comp1;
% comp2=((t>=5)&(t<=6)).*comp2;
% comp3=((t>5) & (t<=8)).*comp3;
% comp4=((t>=2)&(t<=4)).*comp4;
% comp5=((t>=1)&(t<=3)).*comp5;
% comp6=((t>=2)&(t<=6)).*comp6;
% comp7=((t>=1)&(t<=3)).*comp7;
% -----%

% Se muestran las diferentes componentes para la señal
artificial, ordenada
% ascendientemente por la frecuencia:
figure(1)
subplot(8,1,1);
plot(t,comp7,'g')
;title('Señal 7');
xlabel('t(s)');
grid on;

subplot(8,1,2);
plot(t,comp6,'c');
title('Señal 6');
xlabel('t(s)');
grid on;

subplot(8,1,3);
plot(t,comp5,'m');
title('Señal 5');

```



```

xlabel('t(s)');
grid on;

subplot(8,1,4);
plot(t,comp4,'y');
title('Señal 4');
xlabel('t(s)');
grid on;

subplot(8,1,5);
plot(t,comp3,'b');
title('Señal 3');
xlabel('t(s)');
grid on;

subplot(8,1,6);
plot(t,comp2);
title('Señal 2');
xlabel('t(s)');
grid on;

subplot(8,1,7);
plot(t,comp1,'g');
title('Señal 1');
xlabel('t(s)');
grid on;

% Agregado de componentes para obtener la señal
completa
yts = comp1+comp2+comp3+comp4+comp5+comp6+comp7;

% ----- Escenario 6 -----
% ----- %
% SNR = 25;
% yts = awgn(yts,SNR,'measured');
% -----
% ----- %

subplot(8,1,8);
plot(t,yts,'r');
title('Señal Artificial');
xlabel('t(s)');
grid on;

% Presentacion del diagrama tiempo-frecuencia ideal,
mostrando la duracion
% y la frecuencia previamente establecidas.

```

```

figure(2)
hold on
plot (t1,[f1 f1], 'linewidth',2)
plot (t2,[f2 f2], 'linewidth',2)
plot (t3,[f3 f3], 'linewidth',2)
plot (t4,[f4 f4], 'linewidth',2)
plot (t5,[f5 f5], 'linewidth',2)
plot (t6,[f6 f6], 'linewidth',2)
plot (t7,[f7 f7], 'linewidth',2)
grid on
title('DIAGRAMA TIEMPO-FRECUENCIA IDEAL')
xlabel('t(sec)')
ylabel('F(Hz)')
axis([0 duracion 0 fmax*1.1])

legend('Amp.1: 1', 'Amp.2: 1', 'Amp.3: 1', 'Amp.4:
1', 'Amp.5: 1', 'Amp.6: 1', 'Amp.7: 1') % Escenario 1
% legend('Amp.1: 1', 'Amp.2: 1', 'Amp.3: 1', 'Amp.4:
1', 'Amp.5: 1', 'Amp.6: 1', 'Amp.7: 1') % Escenario 2
% legend('Amp.1: 1', 'Amp.2: 1', 'Amp.3: 0.5', 'Amp.4:
0.1', 'Amp.5: 0.2', 'Amp.6: 0.4', 'Amp.7: 0.01') %
Escenario 3
% legend('Amp.1: 1', 'Amp.2: 1', 'Amp.3: 0.5', 'Amp.4:
0.1', 'Amp.5: 0.2', 'Amp.6: 0.4', 'Amp.7: 0.03') %
Escenario 4
% legend('Amp.1: 1', 'Amp.2: 1', 'Amp.3: 1', 'Amp.4:
1', 'Amp.5: 1', 'Amp.6: 1', 'Amp.7: 1') % Escenario 5
% legend('Amp.1: 1', 'Amp.2: 1', 'Amp.3: 1', 'Amp.4:
1', 'Amp.5: 1', 'Amp.6: 1', 'Amp.7: 1') % Escenario 6

xn = yts;

%% 2. ANÁLISIS DOMINIO DE FRECUENCIA:
N=length(xn); % cantidad de muestras discretizadas
FACTOR = 128;
nFFT=2^(ceil(log2(N)))*FACTOR;
f=linspace(0,Fs,nFFT);

%% 2.1 Periodograma con (1-Hann):
window_1=1-hann(length(xn)); % ventana 1-hann
xn_enventanada_1 = window_1 .* xn'; % ventana 1-hann
multiplicada por se;al total
PM1 = 20*log10(abs(fft(xn_enventanada_1,nFFT)));
figure(3)
subplot(2,1,1)

```

```

plot(f,PM1,'k');
grid minor
axis([0 Fs/4 min(PM1) max(PM1)]);% Xmin Xmax Ymin Ymax
title('Periodograma con 1-Hann'); xlabel('f[Hz]');
ylabel('Amplitud')

%% 2.2 Periodograma con Blackmanharris:
window_2=blackmanharris(fix(length(xn))); % ventana
blackmanharris
xn_enventanada_2 = window_2 .* xn'; % ventana
blackmanharris multiplicada por se;al total
PM2 = 20*log10(abs(fft(xn_enventanada_2,nFFT)));
subplot(2,1,2)
plot(f,PM2,'k');
grid minor
axis([0 Fs/4 min(PM2) max(PM2)]);% Xmin Xmax Ymin Ymax
title('Periodograma con Blackmanharris ');
xlabel('f[Hz]'); ylabel('Amplitud')

%% 3. Aplicacion del doble filtro Savisky-Golay

% Doble filtro para 1-Hann
suavizado_1_h=sgolayfilt(PM1',1,2001); % Primer filtro
sgolay para suavizado
suavizado_2_h=sgolayfilt(suavizado_1_h',1,2001); % El
2001 es una ventana

% Doble filtro para Blackmanharris
suavizado_1_b=sgolayfilt(PM2',1,2001); % Filtro sgolay
suavizado_2_b=sgolayfilt(suavizado_1_b',1,2001);

suavizado_total=suavizado_2_h+suavizado_2_b; % suma de
señales suavizadas
figure(4)
plot(f,suavizado_total);
grid on;
axis([0 Fs/4 min(suavizado_total)
max(suavizado_total)]);
title('Señal total suavizada');

%% 4. Resolucion en Tiempo
% 4.1. Blackmanharris. Mas resolucion en amplitud RT
window_RT_B = blackmanharris(fix(length(xn)*0.05)); %
ventana blackmanharris de 5%
noverlap_RT_B=fix(length(window_RT_B)*0.5); % traslape
del 50%

```

```

L_ventana_RT_B=length(window_RT_B);
FACTOR = 128;
nFFT_ventana_RT_B=2^(ceil(log2(L_ventana_RT_B)))*FACTOR;
[P_RT_B,f_RT_B,t_RT_B]=spectrogram(xn,window_RT_B,nove
rlap_RT_B,nFFT_ventana_RT_B,Fs,'yaxis');

% 4.2. 1-Hann. Cambiar el RT por RF (resolucion en
frecuencia)
window_RF_H = 1-hann(fix(length(xn)*0.05)); % ventana
1-hann del 5%
noverlap_RF_H=fix(length(window_RF_H)*0.5); % traslape
del 50%
L_ventana_RF_H=length(window_RF_H);
FACTOR = 128;
nFFT_ventana_RF_H=2^(ceil(log2(L_ventana_RF_H)))*FACTOR;
[P_RF_H,f_RF_H,t_RF_H]=spectrogram(xn,window_RF_H,nove
rlap_RF_H,nFFT_ventana_RF_H,Fs,'yaxis');

%% 5. Tiempo - Frecuecia
P_RT_total = P_RF_H+P_RT_B; %Sumatoria de potencias
encontradas en blackmanharris y 1-hann
% disp('1.- Carga Computacional Metodo Tradicional')
% tic
% figure(5)
% waterfall(f_RF_H,t_RF_H,abs(P_RT_total))
% xlim([0 1000])
% title('Comportamiento en el tiempo y frecuencia
total - Resolucion en Frecuencia');
% xlabel('Frecuencia [Hz]')
% ylabel('Tiempo [s]')
% toc
% figure(6)
%
[P_RF_H,f_RF_H,t_RF_H]=spectrogram(xn,window_RF_H,nove
rlap_RF_H,nFFT_ventana_RF_H,Fs,'yaxis');
%
spectrogram(xn,window_RF_H,noverlap_RF_H,nFFT_ventana_
RF_H,Fs,'yaxis');

% 2da prueba

%% 6. Resolucion en Frecuencia
% 6.1. Blackmanharris. Mas resolucion en tiempo RT

```

```

window_RT_B1 = blackmanharris (fix(length(xn)*0.20)); %
ventana blackmanharris de 20%
noverlap_RT_B1 = fix(length(window_RT_B1)*0.5); %
traslape del 50%
L_ventana_RT_B1 = length(window_RT_B1);
FACTOR = 128;
nFFT_ventana_RT_B1 =
2^(ceil(log2(L_ventana_RT_B1)))*FACTOR;
[P_RT_B1,f_RT_B1,t_RT_B1]=spectrogram(xn,window_RT_B1,
noverlap_RT_B1,nFFT_ventana_RT_B1,Fs,'yaxis');

% 6.2. 1-Hann. Mas resolucion en frecuencia RF
(resolucion en frecuencia)
window_RF_H1 = 1-hann(fix(length(xn)*0.2)); % ventana
1-hann del 20%
noverlap_RF_H1 = fix(length(window_RF_H1)*0.5); %
traslape del 50%
L_ventana_RF_H1 =length(window_RF_H1);
FACTOR = 128;
nFFT_ventana_RF_H1
=2^(ceil(log2(L_ventana_RF_H1)))*FACTOR;

% figure(7)

[P_RF_H1,f_RF_H1,t_RF_H1]=spectrogram(xn,window_RF_H1,
noverlap_RF_H1,nFFT_ventana_RF_H1,Fs,'yaxis');
%
spectrogram(xn,window_RF_H1,noverlap_RF_H1,nFFT_ventan
a_RF_H1,Fs,'yaxis');

%% 7. Tiempo - Frecuecia
% P_RT_totall = P_RF_H1+P_RT_B1; %Sumatoria de
potencias encontradas en blackmanharris y 1-hann
% figure(8)
% waterfall(f_RF_H1,t_RF_H1,abs(P_RT_totall)')
% xlim([0 1000])
% title('Comportamiento en el tiempo y frecuencia
total - Resolucion en Frecuencia');
% xlabel('Frecuencia [Hz]')
% ylabel('Tiempo [s]')

%% 8. Find Peaks aplicado a cada uno de los espectros
(39)
disp('2.- Carga Computacional Metodo Inteligente')
tic

```

```

for j=1:length(Ftot) % Lazo: recorre vector de
frecuencias utilizadas.

    for b=1:length(t_RT_B) % Lazo: recorre todos los
espectros obtenidos.
        x = abs(P_RT_B(:,b)); % Modulo de los
complejos. P_RT_B
        x = x./max(x);

[power,frec]=findpeaks(x,f_RT_B,'MinPeakProminence',0.
1,'MinPeakWidth',0.1,'MinPeakDistance',4);

        for k=1:length(frec) % Lazo: almacenamiento de
picos encontrados en cada espectro
            if abs(Ftot(j)-frec(k))>=0 && abs(Ftot(j)-
frec(k))<4
                mFrecuencia(j,b)=frec(k);
                mAmp(j,b)=power(k);
            end
        end
    end
end
toc
for m=1:length(mFrecuencia(:,1)) % Lazo: encera los
valores del vector que no tienen picos.
    for n=1:length(mFrecuencia)
        if mFrecuencia(m,n)==1
            mFrecuencia(m,n)=0;
        end
    end
end
end

%% 9. Grafica Tiempo - Frecuencia Final

figure(9)
subplot(2,1,1);
for m=1:length(mFrecuencia(:,1)) % Lazo: grafica
continua de cada vector de frecuencias.
    hold on
    grid on
    plot(t_RT_B,mFrecuencia(m,:), '*','LineWidth',3);
    title('DIAGRAMA TIEMPO-FRECUENCIA METODO
INTELIGENTE');
    xlabel('Tiempo [s]')
    ylabel('Frecuencia [Hz]')
end

```

```

axis([0 duracion 1 fmax*1.1])
subplot(2,1,2);
hold on
plot (t1,[f1 f1], 'linewidth',2)
plot (t2,[f2 f2], 'linewidth',2)
plot (t3,[f3 f3], 'linewidth',2)
plot (t4,[f4 f4], 'linewidth',2)
plot (t5,[f5 f5], 'linewidth',2)
plot (t6,[f6 f6], 'linewidth',2)
plot (t7,[f7 f7], 'linewidth',2)
grid on;
title('DIAGRAMA TIEMPO-FRECUENCIA IDEAL')
xlabel('t(sec)')
ylabel('F(Hz)')
axis([0 duracion 0 fmax*1.1])

figure(10)
for m=1:length(mAmp(:,1)) % Lazo: grafica de
amplitudes
    hold on
    grid on
    plot(t_RT_B,mAmp(m,:), '*', 'LineWidth',3);
    title('DIAGRAMA DE AMPLITUDES');
    xlabel('Tiempo [s]')
    ylabel('Amplitudes')
end
axis([0 duracion 0 1.2])

```

ANEXO II

Guía de Uso

La presente Guía tiene el objetivo de explicar la forma adecuada de utilizar el código del ANEXO I. En este código se encuentran los 6 diferentes escenarios antes vistos y los pasos utilizados para poder obtener la carga computacional con métodos tradicionales e inteligentes.

1. En la parte inicial se encuentran las características de los 6 escenarios. Identificar el que se desea analizar.
2. Des comentar la línea de código del apartado "1.1 Frecuencias de las componentes" correspondiente al escenario deseado.
3. Des comentar la línea de código del apartado "1.2 Amplitudes" correspondiente al escenario deseado.
4. Des comentar la línea de código del apartado "1.3 Duraciones de las componentes" correspondiente al escenario deseado.
5. Des comentar la línea de código del apartado "2. Generación de la señal compuesta" correspondiente a las componentes utilizadas para el escenario de análisis.
6. Des comentar la línea de código *legend* correspondiente a la "Figure (2)" para que la gráfica muestre la leyenda perteneciente al escenario por analizar.

NOTA: Para el escenario 6, se debe des comentar las 2 líneas correspondiente a la adición de ruido AWGN.

Para emplear el código completo debe crear un script en el software Matlab versión R2020a. Los valores de carga computacional obtenidos pueden variar dependiendo de las características del computador empleado para la ejecución del código.